1990-06

Transformation of a finite-element model of a piezoelectric spherical shell transducer from a nodal to a spherical Harmonic function representation

McLean, Kathleen Ann

Monterey, California. Naval Postgraduate School

2

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

Transformation of a Finitie-Element Model of a Piezoelectric Spherical Shell Transducer from a Nodal to a Spherical Harmonic Function Representation

by

Kathleen Ann McLean

June 1990

Thesis Co- Advisors: S. R. Baker
D. R. Canright
C. L. Scandrett

UNCLASSIFIED

Security Classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification UNCLASSIFIED	1b Restrictive Markings
2a Security Classification Authority	3 Distribution Availability of Report
2b Declassification/Downgrading Schedule	Approved for public release; distribution is unlimited.
4 Performing Organization Report Number(s)	5 Monitoring Organization Report Number(s)
6a Name of Performing Organization Naval Postgraduate School; 6b Office Symbol (If Applicable) 3A	7a Name of Monitoring Organization Naval Postgraduate School
6c Address (city, state, and ZIP code) Monterey, CA 93943-5000	7b Address (city, state, and ZIP code) Monterey, CA 93943-5000
8a Name of Funding/Sponsoring Organization; 8b Office Symbol	9 Procurement Instrument Identification
8c Address (city, state, and ZIP code)	10 Source of Funding Numbers: Program Element Number; Project No; Task No; Work Unit Accession No

11 Title (Include Security Classification) TRANSFORMATION OF A FINITE-ELEMENT MODEL OF A PIEZOELECTRIC SPHERICAL SHELL TRANSDUCER FROM A NODAL TO A SPHERICAL HARMONIC FUNCTION REPRESENTATION

12 Personal Author(s) McLean, Kathleen A.

13a Type of Report	13b Time Covered	14 Date of Report (year, month,day)	15 Page Count
Master's Thesis	From To	1990 June	66

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms (continue on reverse if necessary and identify by block number)
Field	Group	Subgroup	T-matrix, Finite Element Model, MARTSAM, Sonar Array Modeling

19 Abstract (continue on reverse if necessary and identify by block number)

A new method of array modeling which will be used to predict the performance of low frequency active sonar arrays is being investigated. In support of this effort, a network representation of a spherical shell piezoelectric transducer was developed. The transducer was modeled using the finite-element code MARTSAM, from which a nodal description of the transducer was obtained. A procedure was developed to reduce and transform the nodal description of the transducer into a spherical harmonic description. The spherical harmonic description of the transducer was computed at two frequencies, 112.5 Hz and 1125.3 Hz, corresponding to values of ka of 0.1 and 1.0, respectively where a is the radius of the sphere.

20 Distribution/Availability of Abstract [X] unclassified/unlimited [ ] same as report [ ] DTIC users UNCLASSIFIED	21 Abstract Security Classification UNCLASSIFIED
22a Name of Responsible Individual PROF. S. R. Baker	22b Telephone (Include Area code) (408) 646-2729; 22c Office Symbol 61BA

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted security classification of this page

All other editions are obsolete UNCLASSIFIED

Approved for public release; distribution is unlimited.

Transformation of a Finite-Element Model of a Piezoelectric Spherical Shell Transducer from a Nodal to a Spherical Harmonic Function Representation

by

Kathleen Ann McLean
Lieutenant, United States Navy
B. S., Tufts University, 1980

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN PHYSICS

from the

NAVAL POSTGRADUATE SCHOOL

June 1990

Author: [redacted]

Kathleen A. McLean

Approved by: [redacted]

Steven R. Baker, Co-Advisor

[redacted]

David R. Canright, Co-Advisor

[redacted]

Clyde L. Scandrett, Co-Advisor

[redacted]

Karlheinz E. Woehler, Chairman
Physics Department

ABSTRACT

A new method of array modeling which will be used to predict the performance of low frequency active sonar arrays is being investigated. In support of this effort, a network representation of a spherical shell piezoelectric transducer was developed. The transducer was modeled using the finite-element code MARTSAM, from which a nodal description of the transducer was obtained. A procedure was developed to reduce and transform the nodal description of the transducer into a spherical harmonic description. The spherical harmonic description of the transducer was computed at two frequencies, 112.5 Hz and 1125.3 Hz, corresponding to values of ka of 0.1 and 1.0, respectively, where a is the radius of the sphere.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to thank Professor Steven Baker for teaching me both the joys and frustrations of research work. I truly appreciate his guidance and patience.

Special thanks are due to Professors Canright and Scandrett for providing mathematical expertise, Professor Kuntsal for providing instruction in finite element modeling and assistance with programming, and Dr. Michele McCollum for generously volunteering her time and expertise in modeling.

Finally and most importantly, thanks to my husband, Duncan, for his unceasing encouragement and understanding.

I. INTRODUCTION

The objectives of the research detailed in this thesis are: 1) to produce a nodal description of a spherical shell, piezoelectric transducer using finite-element modeling, and 2) to develop a procedure to transform the nodal description of the transducer into a spherical harmonic description.

This report is organized in three main sections. First, a new method to predict the performance of low frequency active arrays, which provided the motivation for this research, is described. Next, the finite-element modeling and description in terms of spherical harmonics of a particular spherical shell transducer are detailed. Numerical results are included for the transducer operating at 112.5 Hz and 1125.3 Hz, corresponding to values of ka of 0.1 and 1.0, respectively, where a is the radius of the sphere. Lastly, the procedure to reduce and transform the nodal description of the transducer to a spherical harmonic description is described.

II. MODELING OF LOW FREQUENCY ACTIVE ARRAYS

A. BACKGROUND

The trend toward operation of sonar surveillance systems at lower frequencies has necessitated larger, more dense arrays of transducers. This transition has prompted investigation of a new, potentially more efficient, method to predict array performance. This new method of array modeling combines a finite-element representation for each transducer with a mathematical representation of the acoustic field which is equivalent to the T-matrix method which has been applied to elastic scattering problems [Ref. 1]. This approach is explained in greater detail in Appendix A. It has the feature that it accounts for the effects of 'multiple-scattering', which can be quite significant in dense arrays (the distortion of the near-field radiation pattern of a transducer due to a nearby transducer, for example, is a manifestation of multiple-scattering). Applying this method to an array of transducers will ultimately yield a matrix which relates the outgoing acoustic waves to the input electrical excitation of each transducer.

Figure 1 is a schematic diagram of a portion of an array. In this method, the modeling problem is divided into two regions. The term structure refers to the transducer plus some surrounding fluid out to an imaginary, spherical surface, S. The second region is the

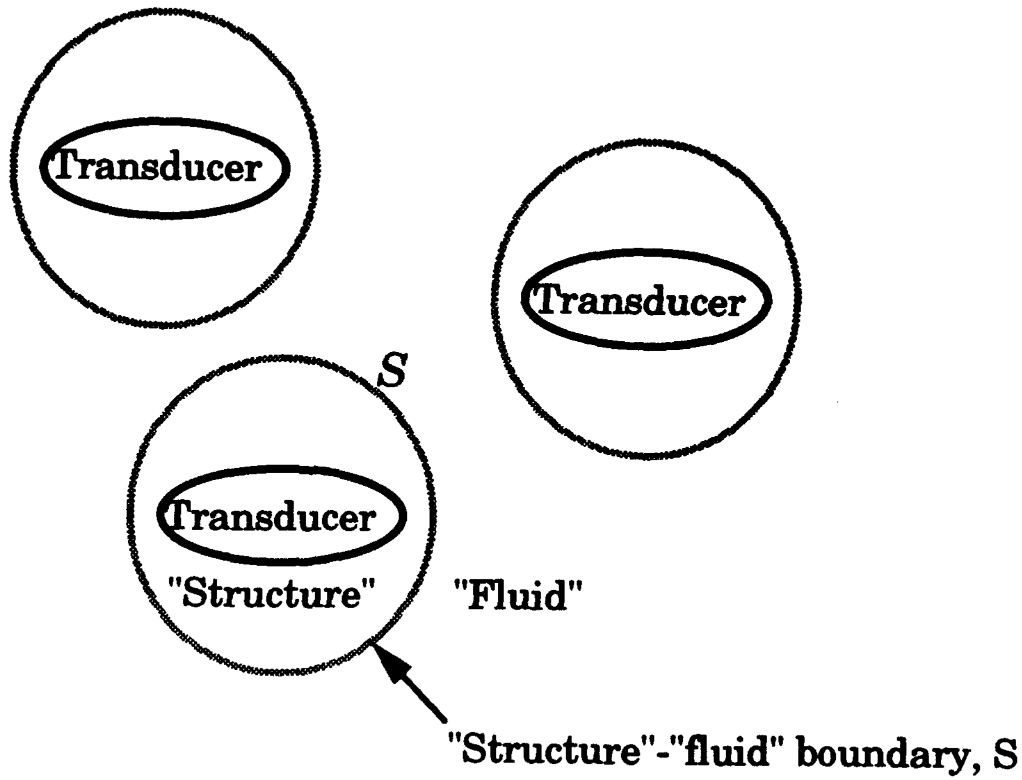

Figure 1: Schematic Diagram of a Portion of a Sonar Array.

fluid outside of S. The array may be composed of any number and/or variety of tranducers. Additionally, it may have any geometry.

The goal is to develop an analytical representation of the structure from a finite-element model and to combine it with an analytical representation of the acoustic field in the fluid to predict the performance of an array of interacting transducers. By separately modeling the structure and the fluid, this approach will simplify the complex task of array modeling. Changing the geometry of an array will require recomputing only the fluid model. Likewise, maintaining the geometry and changing the type of transducer will require a new representation of only the sructure.

B. FLUID REPRESENTATION

The acoustic pressure field in the unbounded fluid is represented as an eigenfunction expansion in terms of spherical waves. It should be noted that all quantities are assumed to vary harmonically with time. This representation is convenient due to the spherical shape of the structures. Figure 2 is the radiation pattern which is produced by three pulsating spherical elastic shells of radius a, with one half wavelength separation and ka equal to 1.0. Each shell is driven with a uniform pressure amplitude on the inside of the surface of the strength indicated by the number inside. The curves shown represent the radiation pattern when the number of spherical harmonics retained in the expansion is varied. The curves which include the first 6, 9, and 11 harmonics are indistinguishable;

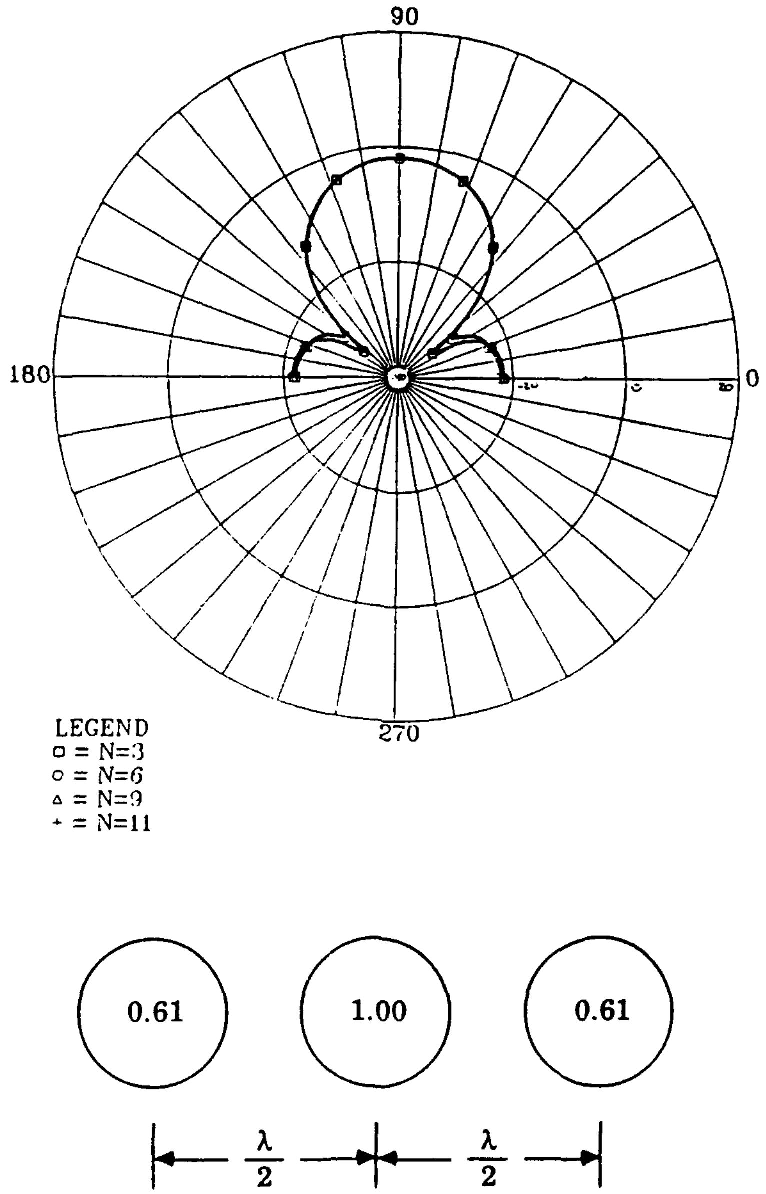

Figure 2: Radiation Pattern of Three Spherical Shells.

therefore, the acoustic field may be accurately described by retaining only about 6 spherical harmonics. [Ref.2]

C. STRUCTURE REPRESENTATION

The structure must be spherical in shape; it may be a transducer with some surrounding fluid or simply a spherical transducer. The structure is first modeled using a finite-element code, which produces a nodal description of the structure. The nodal description is then reduced and transformed into a spherical harmonic description of the structure. This description can then be combined with the spherical harmonic description of the acoustic field on the surface of each sphere to predict the behavior of the array.

III. MODELING OF SPHERICAL SHELL TRANSDUCER

A. FINITE ELEMENT MODEL

A spherical shell piezoelectric transducer was modeled using the finite element code MARTSAM. Figure 3 is the finite element mesh used to model the transducer. The transducer is radially polarized Navy Type 1 ceramic with an outer radius of 7.620 centimeters and thickness of 0.762 centimeters. It is symmetric about the y-axis. Based on this symmetry, it was only necessary to create a two-dimensional model of one half of the transducer. The structure was divided into 60 six-noded triangular elements. This partitioning scheme results in a mesh with 183 nodes. There are two mechanical degrees of freedom associated with each node except the six nodes along the y-axis. Boundary conditions were applied to set the motion of these nodes in the x-direction equal to zero. Consequently, this structure possesses a total of 360 mechanical degrees of freedom. Additional boundary conditions included grounding the electrode on the outer surface of the sphere while maintaining a constant potential on the inner electrode. The structure, therefore, has one electrical degree of freedom.

B. MARTSAM OUTPUT

A modal analysis of the structure was performed by Dr. Michele McCollum of the Naval Research Laboratory, Orlando using the finite-

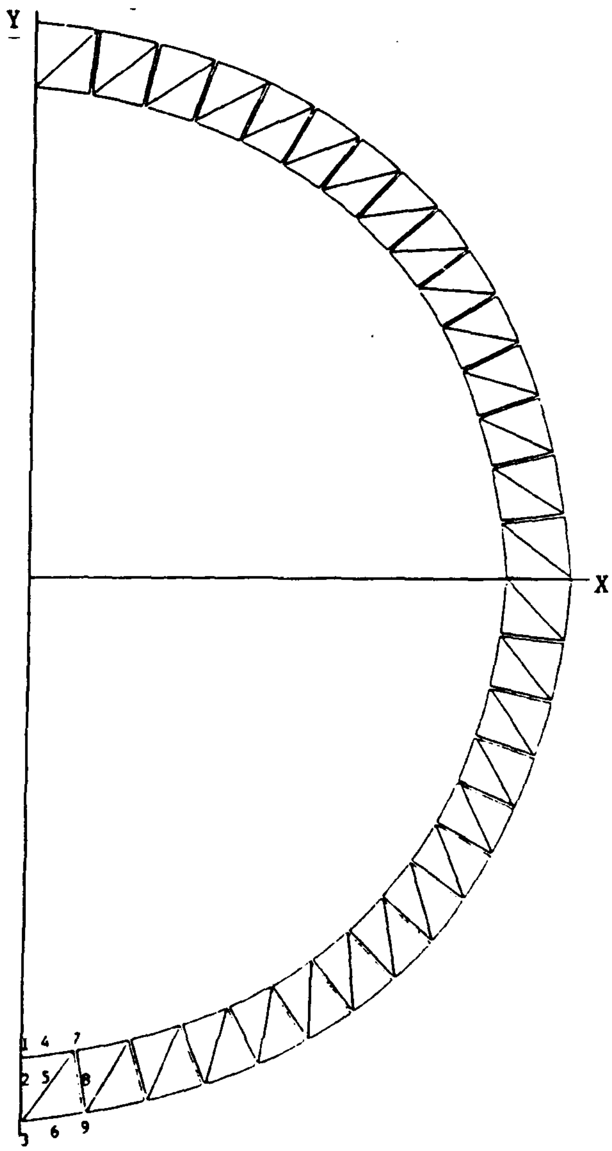

Figure 3: Finite Element Mesh of Spherical Shell Transducer.

element code MARTSAM. For this particular application, the code was used to generate (K_{uu}), (M), and $\underline{K}_{uv}$ which can be substituted into the following set of equations:

$$\begin{pmatrix} (K_{uu}) - \omega^2 (M) & \underline{K}_{uv} \\ \underline{K}_{uv}^T & K_{vv} \end{pmatrix} \begin{pmatrix} \underline{U} \\ V \end{pmatrix} = \begin{pmatrix} \underline{F} \\ Q \end{pmatrix}, \qquad (3.1)$$

where

- $\underline{U}$ and $\underline{F}$ are vectors that contain the nodal values of the displacement field and the applied forces,
- V is the applied electrical potential,
- Q is the electrical charge on the structure,
- (K_{uu}) is the matrix which describes the effect that the stiffness at each node has on all the nodes,
- (M) is the matrix, which describes the effect that the mass at each node has on all the nodes,
- $\underline{K}_{uv}$ is the vector that contains the coupling coefficients that relate the mechanical and electrical degrees of freedom,
- K_{vv} is the capacitance of the transducer for zero displacement everywhere,
- ω is the angular frequency,
- T means transpose,

to completely describe the spherical shell transducer in terms of its nodes at a specified frequency. The matrix of the left side of equation (3.1) has dimension 361 by 361 for this structure.

By means of matrix algebra, which is outlined in Chapter IV, this nodal description of the structure was reduced. The mechanical degrees of freedom associated with the nodes internal to the structure were eliminated as the external force on those nodes is zero. The displacements at the surface nodes were transformed from the Cartesian coordinate system to the polar coordinate system. The degrees of freedom associated with the parallel displacements were also eliminated as external fluid forces are applied normal to the surface. The dimensions of the reduced matrix are 62 by 62, 61 surface normal displacement degrees of freedom and 1 electrical degree of freedom.

C. SPHERICAL HARMONIC DESCRIPTION OF STRUCTURE

Following the procedure detailed in Chapter IV, the reduced nodal description of the transducer was transformed into a spherical harmonic description. Since the finite-element model is restricted to axisymmetric solutions for field quantities, the Legendre polynomials were used for the spherical harmonic functions. Based on the results obtained by Canright and Scandrett [Ref. 2], the first 7 Legendre polynomials were used. The spherical harmonic description of the structure is an 8 by 8 matrix. Figures 4 and 5 are the electrodynamic matrices obtained for the frequencies 112.5 and

1125.3 Hz, respectively, corresponding to values of ka of 0.1 and 1.0, respectively, where a is the radius of the structure. These matrices, $\left(D^{sph}\right)$, are used in the following set of equations:

$$\left(D^{sph}\right)\begin{pmatrix}\underline{U}^{sph}\\ V\end{pmatrix}=\begin{pmatrix}\underline{\sigma}^{sph}\\ Q\end{pmatrix}, \tag{3.2}$$

where $\underline{U}^{sph}$ and $\underline{\sigma}^{sph}$ are vectors that contain the values of the displacement field and stress field in terms of Legendre polynomials (n=0,1,2...6), to describe the transducer. Most of the elements in these matrices are on the order 10^8 - 10^{10}. These numbers represent mechanical interactions while the elements in the last row and column (on the order of 10^{-5} - 10) represent the mechanical and electrical coupling. The element in the last row and column is the (surface only) blocked capacitance.

0.7857444E+10	0.3588145E+09	-0.4129218E+09	0.3586706E+09	-0.4768008E+09	0.3592852E+09	-0.4371310E+09	0.7944787E+01
0.1195338E+09	-0.1196884E+09	0.1196248E+09	-0.1196315E+09	0.1196406E+09	-0.1196303E+09	0.1196340E+09	0.5005859E-05
-0.8258321E+08	0.7179182E+08	0.4257716E+07	0.7177523E+08	-0.7665970E+08	0.7176999E+08	-0.7386322E+08	-0.2953872E-01
0.5127618E+08	-0.5128088E+08	0.5127286E+08	-0.2365810E+07	0.5128466E+08	-0.5474938E+08	0.5126056E+08	0.1061661E-04
-0.5299568E+08	0.3987209E+08	-0.4259016E+08	0.3989635E+08	-0.9552980E+06	0.3991860E+08	-0.4216390E+08	-0.4172413E-02
0.3264719E+08	-0.3262713E+08	0.3262512E+08	-0.3484160E+08	0.3266362E+08	0.4129945E+07	0.3271069E+08	0.4968513E-04
-0.3361179E+08	0.2760376E+08	-0.2840919E+08	0.2760225E+08	-0.2919382E+08	0.2768088E+08	0.1134992E+08	0.1464742E-02
0.3972344E+01	0.1566578E-04	-0.7385326E-01	0.3897119E-04	-0.1877205E-01	0.2759984E-03	0.9518291E-02	0.1363210E-07

Figure 4: Dynamical Matrix in terms of Spherical Harmonics for a Spherical Shell Transducer, f = 112.5.

0.7792194E+10	0.3588350E+09	-0.4131400E+09	0.3589022E+09	-0.4768026E+09	0.3590558E+09	-0.4368015E+09	0.7948672E+01
0.1195407E+09	-0.1394371E+09	0.1196574E+09	-0.1193090E+09	0.1196429E+09	-0.1191066E+09	0.1196193E+09	-0.6585382E-05
-0.8261550E+08	0.7178045E+08	0.9185140E+06	0.7178029E+08	-0.7661498E+08	0.7178077E+08	-0.7384097E+08	-0.2919376E-01
0.5127371E+08	-0.5112783E+08	0.5126926E+08	-0.3829749E+07	0.5128490E+08	-0.5475942E+08	0.5126356E+08	0.1129135E-04
-0.5298319E+08	0.3987425E+08	-0.4256295E+08	0.3989227E+08	-0.1805442E+07	0.3991801E+08	-0.4216915E+08	-0.4190601E-02
0.3264763E+08	-0.3248068E+08	0.3262625E+08	-0.3484698E+08	0.3266027E+08	0.3566874E+07	0.3271029E+08	0.5073217E-04
-0.3359650E+08	0.2760695E+08	-0.2840250E+08	0.2760115E+08	-0.2919360E+08	0.2767997E+08	0.1095424E+08	0.1452875E-02
0.3974318E+01	0.3907830E-05	-0.7297906E-01	0.4182942E-04	-0.1885402E-01	0.2665687E-03	0.9438062E-02	0.1363104E-07

Figure 5: Dynamical Matrix in terms of Spherical Harmonics for a Spherical Shell Transducer, f = 1125.3.

IV. TRANSFORMATION PROCEDURE

A. REDUCED NODAL DESCRIPTION

Matrix equation (3.1), for simplicity, can be represented by

$$(D^{nod})\,\underline{U}^{nod} = \underline{F}^{nod}, \tag{4.1}$$

where

- $\left(D^{nod}\right)$ is the square dynamical matrix,
- $\underline{U}^{nod}$ is the vector containing the displacement field and the electrical potential, and
- $\underline{F}^{nod}$ is the vector containing the applied force field and the electrical charge.

The dimensions of the dynamical matrix are quite large for even the simplest transducer. In modeling the fluid-loaded structure, however, the size of this description can be greatly reduced by applying matrix algebra as the following procedure describes. In performing this reduction, the accuracy of the description is not diminished since the fluid forces act only at the surface bounding the structure.

1. Forces and Internal Nodes

$\underline{F}^{nod}$ is initially a column vector whose entries are the force applied to each node listed by ascending node number and direction followed by the electrical charge. However, the applied force is zero at all the nodes that are internal to the structure as those nodes are inaccessible to the surrounding fluid. $\underline{F}^{nod}$ is rearranged so that all the zero entries come first. When $\underline{F}^{nod}$, the vector on the right hand side of the equation, is reordered, the rows of the dynamical matrix on the left side of the equation must be rearranged accordingly. $\underline{U}^{nod}$ must also be reordered in the same manner as $\underline{F}^{nod}$ which necessitates a corresponding rearranging of the columns of the dynamical matrix.

2. Transformation of Coordinate System

MARTSAM describes the transducer in terms of Cartesian coordinates. A transformation matrix, (T), can be defined for each surface node such that it performs the following operations:

$$\begin{pmatrix} \underline{U}_x \\ \underline{U}_y \end{pmatrix} = (T) \begin{pmatrix} \underline{U}_{perp} \\ \underline{U}_{par} \end{pmatrix} \quad \text{and}$$

$$\begin{pmatrix} \underline{F}_x \\ \underline{F}_y \end{pmatrix} = (T) \begin{pmatrix} \underline{F}_{perp} \\ \underline{F}_{par} \end{pmatrix}, \tag{4.2}$$

where, respectively,

- $\underline{U}_x$ and $\underline{U}_y$ are the displacements in the x- and y-directions,
- $\underline{F}_x$ and $\underline{F}_y$ are the applied forces in the x- and y-directions,
- $\underline{U}_{perp}$ and $\underline{U}_{par}$ are the displacements in the directions perpendicular and parallel to the surface, and

- $\underline{F}_{perp}$ and $\underline{F}_{par}$ are the applied forces in the directions perpendicular and parallel to the surface.

For the mesh shown in Figure 3, for a single element,

$$(T) = \begin{pmatrix} \sin\theta & -\cos\theta \\ \cos\theta & \sin\theta \end{pmatrix} = 1/R \begin{pmatrix} X & -Y \\ Y & X \end{pmatrix}, \qquad (4.3)$$

where

- θ is the angle each surface node makes with the positive y-axis,
- X and Y are the Cartesian coordinates of each node, and
- R is the radially distance of each node from the center of the sphere.

Substituting equations (4.2) into equation (4.1) yields

$$(D^{nod})\,(T)\,\underline{U}^{polar} = (T)\,\underline{F}^{polar}, \qquad (4.4)$$

where the vectors U and F are now expressed in terms of polar coordinates. Multiplying each side of equation (4.4) by the inverse of (T) gives

$$(T)^{-1}\,(D^{nod})\,(T)\,\underline{U}^{polar} = \underline{F}^{polar}, \qquad (4.5)$$

where $(T)^{-1}(D^{nod})(T)$ represents the dynamical matrix in terms of nodes and the polar coordinate system.

3. **Forces and Parallel Direction**

The elements of $\underline{F}^{polar}$ are arranged as follows: the forces applied to the internal nodes, the forces applied in the directions perpendicular to the transducer at the surface nodes and parallel to the transducer at the surface nodes, and the electrical charge. However, since only the forces applied normal to the transducer are non-zero, $\underline{F}_{par}$ is the null vector. Using the procedure described in section 1, $\underline{F}^{polar}$ and $\underline{U}^{polar}$ must be reordered such that $\underline{F}_{par}$ and $\underline{U}_{par}$ precede $\underline{F}_{perp}$ and $\underline{U}_{perp}$. The rows and columns of the dynamical matrix must be rearranged accordingly.

4. **Reduction of Dynamical Matrix**

The procedures in sections 1, 2 and 3 have arranged the dynamical matrix into the proper form to be reduced using matrix algebra. The matrix equation is now in the form

$$\begin{pmatrix} (UL) & (UR) \\ (LL) & (LR) \end{pmatrix} \begin{pmatrix} \underline{U}_o \\ \underline{U}_{perp} \end{pmatrix} = \begin{pmatrix} \underline{0} \\ \underline{F}_{perp} \end{pmatrix}, \tag{4.6}$$

where

- $(UL), (UR), (LL), (LR)$ are subdivisions of the dynamical matrix,
- $\underline{U}_o$ is the displacements at the internal nodes and in the direction parallel to the surface nodes,

- $\underline{U}_{perp}$ is the normal displacements and the electrical potential,
- $\underline{0}$ is the null vector, and
- $\underline{F}_{perp}$ are the forces applied normal to the surface and the electrical charge.

The corresponding simultaneous equations are

$$(UL)\,\underline{U}_o + (UR)\,\underline{U}_{perp} = \underline{0}\text{ , and} \tag{4.7}$$

$$(LL)\,\underline{U}_o + (LR)\,\underline{U}_{perp} = \underline{F}_{perp}\,. \tag{4.8}$$

Solving equation (4.7) for $\underline{U}_o$ yields

$$\underline{U}_o = -\ (UL)^{-1}\,(UR)\,\underline{U}_{perp}\,. \tag{4.9}$$

Substituting equation (4.9) into equation (4.8) gives

$$\Big((LR) - (LL)\,(UL)^{-1}\,(UR)\Big)\,\underline{U}_{perp} = \underline{F}_{perp}\,, \tag{4.10}$$

where $\Big((LR) - (LL)\,(UL)^{-1}\,(UR)\Big)$ is the reduced dynamical matrix describing nodal interactions. This matrix is square, and its dimension is the number of surface nodes plus the electrical degree of freedom. The computer code, which computes the reduced dynamical matrix for the spherical shell transducer described in Chapter III, is listed in Appendix B [Ref. 3:p. 38].

B. TRANSFORMATION TO SPHERICAL HARMONIC DESCRIPTION

The reduced nodal description of the transducer given by equation (4.10) can be written as

$$\begin{pmatrix} \left(D_{uu}^{nod}\right) & \underline{K}^{nod} \\ \underline{K}^{nod\ T} & C_{EB} \end{pmatrix} \begin{pmatrix} \underline{U}^{nod} \\ V \end{pmatrix} = \begin{pmatrix} \underline{F}^{nod} \\ Q \end{pmatrix}, \tag{4.11}$$

where

- $\left(D_{uu}^{nod}\right)$ is the reduced $\left(K_{uu}\right) - \omega^2 \left(M\right)$ matrix,
- $\underline{K}^{nod}$ is the reduced vector containing the coupling coefficients that relate the mechanical and electrical degrees of freedom,
- C_{EB} is the (surface only) blocked capacitance,
- $\underline{U}^{nod}$ is the vector containing normal displacements,
- V is the electrical potential,
- $\underline{F}^{nod}$ is the vector containing the forces applied normal to the surface,
- Q is the electrical charge, and
- T means transpose.

A spherical harmonic description of the transducer can be obtained, in the form

$$\left(D^{sph}\right) \underline{U}^{sph} = \underline{\sigma}^{sph} , \qquad (4.12)$$

by performing the following tranformations.

1. Transformation of $\underline{U}^{nod}$ to $\underline{U}^{sph}$

The surface normal displacement distribution can be written as

$$U(\theta) = \sum_{n}^{N} U_n^{nod} N_n(\theta) , \qquad (4.13)$$

where $U_n^{nod} = U(\theta = \theta_n)$, and $N(\theta)$ is the finite-element interpolation function where $N_n(\theta) = \delta(\theta = \theta_n)$ [Ref. 4]. Since the finite-element model created to represent the transducer in this investigation is restricted to axisymmetric solutions for all field quantities, the Legendre polynomials were chosen for the set of harmonic functions. The surface normal displacement distribution can then be expressed approximately by a truncated series as

$$U(\theta) = \sum_{m}^{M} U_m^{sph} P_m(\cos\theta) , \qquad (4.14)$$

where $U_m^{sph} = (1/C_m) \int_0^{\pi} U(\theta) P_m(\cos\theta) \sin\theta \, d\theta$ and $C_m = \int (P_m)^2 \sin\theta \, dS$, the usual normalization constant associated with P_m. For the

Legendre polynomials, $C_m = (2l+1)/2$ where l is the order. From equations (4.13) and (4.14), it follows that

$$U_n^{nod} = \sum_m^M U_m^{sph} \, P_m(\cos\theta = \cos\theta_n) . \qquad (4.15)$$

Equation (4.15) expressed in matrix form is

$$\underline{U}^{nod} = \left(P\right) \underline{U}^{sph} , \qquad (4.16)$$

where the matrix elements $P_{nm} = P_m(\cos\theta = \cos\theta_n)$.

2. Transformation of $\underline{F}^{nod}$ to $\underline{\sigma}^{sph}$

The self-consistent force at node n due to a distributed stress σ is given by

$$F_n^{nod} = \int \sigma N_n dS, \qquad (4.17)$$

where the integration is performed over the surface of the structure. The distributed stress can be expanded similarly to the displacement as

$$\sigma(\theta) = \sum_m^M \sigma_m^{sph} \, P_m(\cos\theta) . \qquad (4.18)$$

The dynamical equation describing the transducer in terms of $\underline{\sigma}^{sph}$ follows from the equation for the coefficients of the expansion of the stress field in the Legendre polynomials:

$$\sigma_m^{sph} = (1/C_m) \int \sigma \, P_m \sin\theta \, dS. \tag{4.19}$$

The expansion,

$$P_m(\theta) = \sum_{n}^{N} P_{nm} N_n , \tag{4.20}$$

where $P_{nm} = P_m(\theta_n)$, can be made. Substitution of equation (4.20) into equation (4.19) gives

$$\sigma_m^{sph} = (1/C_m) \sum_{n}^{N} \left(\int \sigma N_n dS\right) P_{mn} . \tag{4.21}$$

Using equation (4.17), equation (4.21) becomes

$$\sigma_m^{sph} = (1/C_m) \sum_{n}^{N} F_n^{nod} P_{mn} . \tag{4.22}$$

In matrix form, equation (4.22) can be written

$$\underline{\sigma}^{sph} = \left(C\right)^{-1} \left(P\right)^{T} \underline{F}^{nod} , \tag{4.23}$$

where

- $(C) = \mathrm{diag}\,(C_m)$,
- $(P)_{nm} = P_m(\cos\theta_n)$, and
- $F_n^{nod} = \int \sigma N_n \, dS$.

Equation (4.23) is the desired expression relating $\underline{\sigma}^{sph}$ to $\underline{F}^{nod}$.

Matrix equation (4.11) can be expressed as

$$\left(D_{uu}^{nod}\right) \underline{U}^{nod} + \underline{K}^{nod} V = \underline{F}^{nod}, \text{ and} \tag{4.24}$$

$$\underline{K}^{nod\,T} \underline{U}^{nod} + C_{EB} V = Q . \tag{4.25}$$

Substitution of equation (4.24) into equation (4.23) gives

$$\underline{\sigma}^{sph} = (C)^{-1} (P)^T \left(D_{uu}^{nod}\right) \underline{U}^{nod} + (C)^{-1} (P)^T \underline{K}^{nod} V . \tag{4.26}$$

Therefore, by substituting equation (4.16), equation (4.26) reduces to

$$\left(D_{uu}^{sph}\right) \underline{U}^{sph} + \underline{K}^{sph} V = \underline{\sigma}^{sph} , \tag{4.27}$$

where $\left(D_{uu}^{sph}\right) = (C)^{-1} (P)^T \left(D_{uu}^{nod}\right) (P)$ and $\underline{K}^{sph} = (C)^{-1} (P)^T \underline{K}^{nod}$.
It follows from equations (4.16) and (4.25) that

$$\underline{K}^{nod\,T} (P)\, \underline{U}^{sph} + C_{EB} V = Q . \tag{4.28}$$

Hence the reduced dynamical matrix equation in terms of Legendre polynomials coefficient degrees of freedom is

$$\begin{pmatrix} \left(D_{uu}^{sph}\right) & (C)^{-1}\,(P)^T \underline{K}^{nod} \\ \underline{K}^{nod\,T}\,(P) & C_{EB} \end{pmatrix} \begin{pmatrix} \underline{U}^{sph} \\ V \end{pmatrix} = \begin{pmatrix} \underline{\sigma}^{sph} \\ Q \end{pmatrix}. \tag{4.29}$$

The computer code, which computes the transformation from the reduced nodal dynamical matrix to the spherical harmonic matrix, is listed in Appendix C [Ref. 3:p. 38].

V. CONCLUSIONS AND RECOMMENDATIONS

This thesis represents the first step in an investigation which, when successfully completed, will significantly improve the Navy's ability to predict the performance of dense sonar arrays. A network (matrix) representation of a spherical shell transducer has been calculated in terms of spherical harmonics. The procedure for reducing and transforming a nodal description of a transducer into a spherical harmonic description has been documented.

For this thesis, a two-dimensional model of a spherical shell transducer was created and a procedure was developed to reduced and transform its nodal description into a spherical harmonic one, it is recommended that a three-dimensional model be created and a similar transformation procedure be developed. Since it is desirable that ω remain a free parameter when describing a transducer, it is suggested that the procedure detailed in Chapter IV be rederived following Benthien's method to obtain the dynamical matrix as a function of ω [Ref. 5]. It is also recommended that experimental data be obtained to verify the matrix representation developed in this thesis.

APPENDIX A:
EXCERPTS FROM PROPOSAL FOR LFA ARRAY MODELING

APPLICATION OF THE T-MATRIX METHOD TO LOW-FREQUENCY ACTIVE ARRAY PERFORMANCE PREDICTION

TECHNICAL PROPOSAL

A. Introduction and Related Work

The direction of active sonar surveillance systems is toward lower frequencies, requiring arrays of large, high power transducers. The successful design and operation of such arrays requires the ability to reliably predict their performance.

Array performance prediction is a coupled structure-fluid problem [1]. For the analysis of complex structures, such as a sonar transducer, the finite-element method (FEM) is preeminent [2]. Many FEM computer codes exist, some of which include piezoelectric elements for the analysis of piezoelectric transducers, e.g. MARTSAM, ATILA. For application to coupled structure-fluid problems, the major drawback of the finite-element method is the difficulty of modeling the infinite fluid. So-called boundary elements, derived from a Helmholtz Integral formulation of the exterior radiation problem, have been introduced into the FEM to terminate the fluid mesh [3,4,5]. These elements typically impose an outgoing radiation impedance condition on the adjacent fluid element. The frequency dependence of the radiation impedance is commonly approximated by an interpolation between the low- and high-frequency asymptotic limits [5]. The application of a combined FEM-boundary integral method to array performance prediction has not appeared in the literature. A full Helmholtz Integral formulation can be applied at the bounding surface [2,3], but this technique is limited to small problems by the number of degrees of freedom, and so is not practical to model an entire array. Hence there is a need for economical methods for array performance prediction. This proposal addresses that need.

The method we propose is formulated in terms of coupled networks, one for each radiating element and one for the acoustic field. It is equivalent to the T-matrix method which has been applied to scattering problems [7,8,9,10], in that ultimately a matrix can be derived which relates the outgoing acoustic wave to the input electrical excitation of each transducer (the equivalent reciprocal problem is to relate the.electrical output of each transducer to an incident wave). One could in fact consider the proposed research the application of the T-matrix method to sonar array performance prediction. As in the T-matrix method, we adopt the idea of representing the acoustic field as an eigenfunction expansion and solving for the coefficients self-consistently. No restriction is placed on the arrangement of the radiators, and so multiple-scattering of all orders is rigorously included. In addition, we plan to explicitly take into account the dynamical properties of the radiators. These are to be found for a rea' transducer using a finite-element computer code [2,13].

B. Proposed Effort

Goal

The goal of the proposed research is to produce the most economical yet complete description possible of sonar array performance, with specific application to low frequency active arrays. The idea is to combine an analytical representation of a single element (derived from a finite-element model for a real transducer) with an analytical representation of the acoustic field to predict the performance of an array of interacting transducers. Especially for the usual case of an array of identical elements, since the dynamic behavior of only a single element need be computed, and since this computation need be done only once, regardless of array geometry, a variety of operating frequencies and array geometries may be analyzed most economically.

Method

A schematic diagram of a portion of an array of sonar transducers is shown below.

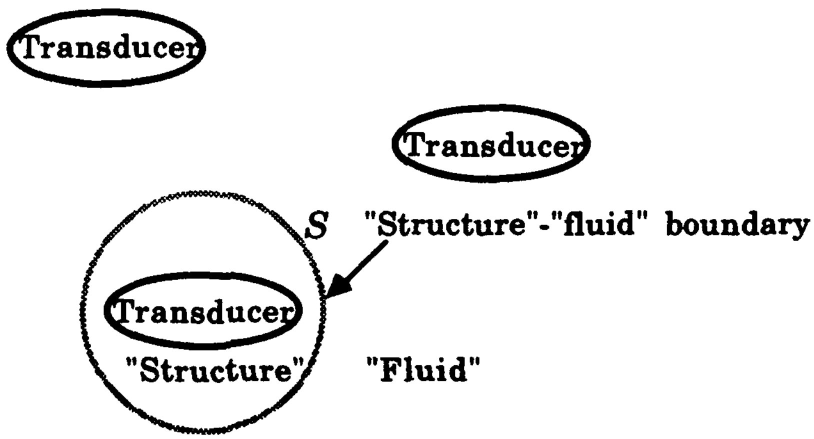

Figure 1. Schematic of a portion of an array of sonar transducers.

The transducers need not be identical, nor in any particular arrangement. There may or may not be an acoustic wave incident from a source external to the array. The closed surface labeled S indicates the boundary between "structure" and "fluid". S may coincide with the physical boundary of a transducer, or it may include some surrounding fluid.

There are several reasons to choose S to lie within the fluid surrounding a transducer. The most important is that, if the "structure"-"fluid" boundary coincides with a constant-coordinate surface of a coordinate system in which the wave equation is separable, then the eigenfunctions of the wave equation in that coordinate system are a particularly convenient choice as the basis for expanding both the surface normal velocity on S and the acoustic field within the "fluid". This makes it much easier, for example, to compute the self- and mutual radiation impedances. Also, by including some surrounding fluid as part of the "structure", the burden of carrying many higher-order (higher spatial frequency) terms in an eigenfunction expansion of the acoustic field, which might be required to describe the local flow field near a real transducer, would be relieved. These degrees of freedom would be considered as internal to the structure, with the result that far fewer degrees of freedom would be required to describe the acoustic field. This is important because only the description of the acoustic field within the "fluid" needs to be recalculated for a change in array geometry.

Conversely, by including some surrounding fluid, the description of the acoustic field would be unchanged by a change in the physical structure, for example, by exchanging one type of transducer for another (e.g. flextensional for hydroacoustic). The "structure" could even be composed of more than one physical transducer. This would be useful, for example, to analyze the performance of an array composed of groups of closely-spaced transducers.

Including some surrounding fluid as part of the "structure" raises the possibility of introducing spurious structural resonances, which may cause problems for a numerical computation [10]. The consequences of this will have to be investigated.

It is convenient to discuss the array performance problem further in terms of a coupled network representation. The diagram below depicts the coupling between "structure" and "fluid" for one surface normal velocity degree-of-freedom. For simplicity, the transducer is considered to be reciprocal.

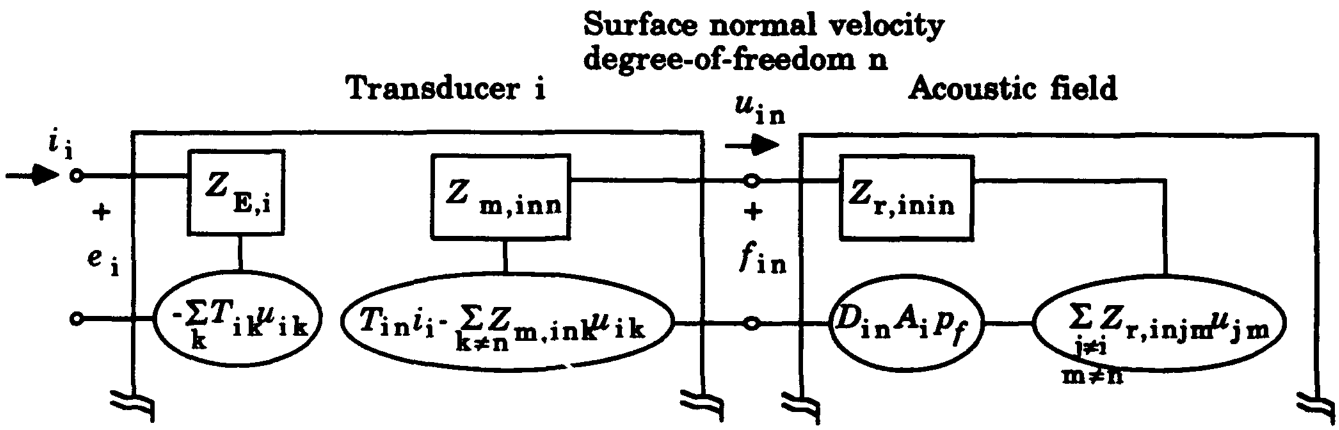

Figure 2. Coupled reciprocal network representation of structure-fluid interaction for one surface normal velocity degree-of-freedom.

The network equations representing transducer i and the acoustic field are

$$e_i = Z_{E,i} i_i - \sum_k T_{ik} u_{ik}, \qquad \text{(transducer i)}$$

$$f_{in} = T_{in} i_i - \sum_k Z_{m,ink} u_{ik},$$

$$f_{in} = D_{in} A_i p_f + \sum_{j,m} Z_{r,injm} u_{jm}. \qquad \text{(acoustic field)}$$

The subscripts i and j refer to particular transducers, and the subscripts k, n and m refer to particular surface normal velocity degrees-of-freedom (DOF) on a "structure"-"fluid" boundary, assumed to be a constant-coordinate surface of a coordinate system in which the wave equation is separable. The meaning of the remaining symbols is

e_i = the voltage across the electrical terminals of transducer i,
i_i = the current through the electrical terminals of transducer i,
f_{in} = the modal force amplitude of transducer i, DOF n,
u_{in} = the modal surface normal velocity amplitude of transducer i, DOF n,
$Z_{E,i}$ = the blocked electrical impedance of transducer i,

$Z_{m,inn}$ = the open-circuit self mechanical impedance of transducer i, DOF n,
$Z_{m,inm}$ = the open-circuit mutual mechanical impedance between transducer i, DOF n, and DOF m
T_{in} = the transduction coefficient of transducer i, DOF n,
$Z_{r,inin}$ = the self radiation impedance of transducer i, DOF n,
$Z_{r,injm}$ = the mutual radiation impedance between transducer i, DOF n, and transducer j, DOF m,
p_f = the incident free-field pressure, if any (usually assumed to be a plane wave),
A_i = the surface area of the "structure"-"fluid" boundary of transducer i,
D_{in} = the diffraction constant of transducer i, DOF n (the diffraction constant equals the ratio of the blocked modal force amplitude per unit area to the incident free-field pressure).

The acoustic field is to be represented as the superposition of the acoustic field due to each radiator and an optional incident field. The surface normal velocity and the acoustic field of each radiating element are to be represented as expansions in the free-space eigenfunctions of the wave equation. An addition theorem [8,11,12] is to be used to express the field due to one radiator at the surface of another. The $Z_{r,injm}$ and D_{in} are to be found in terms of the eigenfunction expansions by applying the boundary condition at the surface of each radiator that $u_{jm}=\delta_{ij}\delta_{mn}$ and $u_{jm}=0$, respectively. No restriction is placed on the arrangement of the radiators, and so multiple-scattering of all orders is rigorously included. A self-consistent solution for the coefficients of the eigenfunction expansions for each radiator is to be found by applying an impedance-matching boundary condition at the surface of each radiator. The mechanical impedance of each radiator is to be represented in terms of the modal parameters of its in-vacuo eigenfunctions. These are to be found for a real transducer using a finite-element computer code [2,13].

D. References

1. Junger, M. C., and Feit, D., Sound, Structures, and their Interaction, Second Edition, MIT Press, Cambridge, MA, 1986.
2. Mathews, I. C., "Numerical techniques for three-dimensional steady-state fluid-structure interaction.", J. Acoust. Soc. Am. 79, 1317-1325 (1986).
3. Cunefare, K. A., and Koopman, G. A., "A boundary element for acoustic radiation valid for all wavenumbers.", J. Acoust. Soc. Am. 85, 39-48 (1989).
4. DiMaggio, F. L., Doubly asymptotic approximations as absorbing fluid boundaries.", Mech. Res. Commun. 8, 349-354 (1981).
5. Geers, T. L., "Doubly asymptotic approximations for transient motions of submerged structures.", J. Acoust. Soc. Am. 64, 1500-1508 (1978).
6. Wilton, D. T., "Acoustic radiation and scattering from elastic structures.", International J. Numer. Meth. Eng. 13, 123-138 (1978).
7. Bostrom, A., "Scattering of stationary acoustic waves by an elastic obstacle immersed in a fluid.", J. Acoust. Soc. Am. 67, 390-398 (1980).
8. Peterson, B., and Strom, S., "Matrix formulation of acoustic scattering from an arbitrary number of scatterers.", J. Acoust. Soc. Am. 56, 771-780 (1974).
9. Varadan, V. V., and Varadan, V. K., "Configurations with finite numbers of scatterers -- A self-consistent T-matrix approach.", J. Acoust. Soc. Am. 70, 213-217 (1981).
10. Shaw, R. P., Fukui, T., and Wang, H. C., "Boundary integral equation/element methods in computational acoustics.", in Computational Acoustics: Wave Propagation, D. Lee, R. H. Sternberg, M. H. Schultz (Editors), Elsevier Science Publishers B. V. (North Holland), 1988, pp. 157-176.
11. Friedman, B., and Russek, J., "Addition theorem for spherical waves.", Quart. Appl. Math. 12, 13-23 (1954).
12. King, B. J., and Van Buren, A. L., "A general addition theorem for spheroidal wave functions.", SIAM J. Math. Anal. 4, 149-160 (1973).
13. DeRuntz, J. A., and Geers, T. L., "Added mass computation by the boundary integral method.", Int. J. Num. Methods Eng. 12, 531-549 (1977).

APPENDIX B:

```
c       PROGRAM REDDYNMTRX:  computes the reduced nodal dynamical matrix
c       for LT McLean's radially polarized piezoelectric spherical shell
c
        real omega,omega2,pi,freq,rr,pint(361,361)
        real dyn(361,361),kuu(360,360),mas(360,360)
        real r(361,361),g(361,361)
        real a(299,299),b(299,62),c(62,299),d(62,62)
        real nn(62,62),sum
        real p(62,299),pp(62,62),prod(361,361)
        real s(361,361),mat(361,361),y(299,299),indx(299)
        real t(361,361),tt(361,361)
        real phtoh(8,8)
        real ccos(59),ssin(59),c2(59)
c
c       OLD DATA FILES
        open(unit=5,file='transform.sph',status='old')
        open(unit=8,file='kuu.sph',status='old')
        open(unit=9,file='mas.sph',status='old')
        open(unit=10,file='kue.sph',status='old')
        open(unit=11,file='kuel.sph',status='old')
c
c       NEW FILE FOR STORING THE REDUCED DYNAMICAL MATRIX
        open(unit=4,file='reddynmtrx.dat',status='new',form='unformatted')
c
        pi=acos(-1.0)
c
c       ASSEMBLE THE FULL DYNAMICAL MATRIX
c
c       Read in Kuu(360,360)
        do i=1,360
           do j=1,360,5
                read(8,*)(kuu(i,k), k=j,j+4)
           end do
        end do
c
c       Read in Muu(360,360)
        do i=1,360
           do j=1,360,5
                read(9,*)(mas(i,k), k=j,j+4)
           end do
        end do
c
c       Read Kue(360,1) into the 361th column of dyn(361,361)
        do i=1,360
           read(10,*)dyn(i,361)
        end do
c
c       Read Kue transpose (1,360) into the 361th row of dyn(361,361)
        do j=1,360
           read(11,*)dyn(361,j)
        end do
c
c       dyn(361,361)=capacitance for zero displacement everywhere
        dyn(361,361)=1.8515495E-08
c
c       ASSEMBLE K-w**2M MATRIX
        write(*,*)'Enter frequency in Hz:'
        read(*,*)freq
        omega=(2.0*pi*freq)
        omega2=omega**2
```

```
      do i=1,360
         do j=1,360
             dyn(i,j)=kuu(i,j)-omega2*mas(i,j)
         end do
      end do
c
c     TEST SYMMETRY OF ASSEMBLED FULL DYNAMICAL MATRIX
c     nerr=0
c     do i=1,361
c        do j=i,361
c           if (dyn(j,i).ne.dyn(i,j)) then
c              write (*,*)'Symmetry error in full dyn. matrix at (i,j)=',i,j
c              nerr=nerr+1
c           end if
c        end do
c     end do
c     write (*,*) 'Number of symmetry errors in full dynamical matrix =',nerr
c     if (nerr.gt.0) then
c        STOP
c     end if
c
c     REARRANGE ROWS SO ZERO (INTERNAL) FORCES COME FIRST
      do j=1,361
      r(1,j)=dyn(1,j)
      r(2,j)=dyn(2,j)
      r(241,j)=dyn(3,j)
      r(3,j)=dyn(4,j)
      r(4,j)=dyn(5,j)
      r(5,j)=dyn(6,j)
      r(6,j)=dyn(7,j)
      r(242,j)=dyn(8,j)
      r(243,j)=dyn(9,j)
      r(7,j)=dyn(10,j)
      r(8,j)=dyn(11,j)
      r(9,j)=dyn(12,j)
      r(10,j)=dyn(13,j)
      r(244,j)=dyn(14,j)
      r(245,j)=dyn(15,j)
      r(11,j)=dyn(16,j)
      r(12,j)=dyn(17,j)
      r(13,j)=dyn(18,j)
      r(14,j)=dyn(19,j)
      r(246,j)=dyn(20,j)
      r(247,j)=dyn(21,j)
      r(15,j)=dyn(22,j)
      r(16,j)=dyn(23,j)
      r(17,j)=dyn(24,j)
      r(18,j)=dyn(25,j)
      r(248,j)=dyn(26,j)
      r(249,j)=dyn(27,j)
      r(19,j)=dyn(28,j)
      r(20,j)=dyn(29,j)
      r(21,j)=dyn(30,j)
      r(22,j)=dyn(31,j)
      r(250,j)=dyn(32,j)
      r(251,j)=dyn(33,j)
      r(23,j)=dyn(34,j)
      r(24,j)=dyn(35,j)
      r(25,j)=dyn(36,j)
      r(26,j)=dyn(37,j)
```

```
r(252,j)=dyn(38,j)
r(253,j)=dyn(39,j)
r(27,j)=dyn(40,j)
r(28,j)=dyn(41,j)
r(29,j)=dyn(42,j)
r(30,j)=dyn(43,j)
r(254,j)=dyn(44,j)
r(255,j)=dyn(45,j)
r(31,j)=dyn(46,j)
r(32,j)=dyn(47,j)
r(33,j)=dyn(48,j)
r(34,j)=dyn(49,j)
r(256,j)=dyn(50,j)
r(257,j)=dyn(51,j)
r(35,j)=dyn(52,j)
r(36,j)=dyn(53,j)
r(37,j)=dyn(54,j)
r(38,j)=dyn(55,j)
r(258,j)=dyn(56,j)
r(259,j)=dyn(57,j)
r(39,j)=dyn(58,j)
r(40,j)=dyn(59,j)
r(41,j)=dyn(60,j)
r(42,j)=dyn(61,j)
r(260,j)=dyn(62,j)
r(261,j)=dyn(63,j)
r(43,j)=dyn(64,j)
r(44,j)=dyn(65,j)
r(45,j)=dyn(66,j)
r(46,j)=dyn(67,j)
r(262,j)=dyn(68,j)
r(263,j)=dyn(69,j)
r(47,j)=dyn(70,j)
r(48,j)=dyn(71,j)
r(49,j)=dyn(72,j)
r(50,j)=dyn(73,j)
r(264,j)=dyn(74,j)
r(265,j)=dyn(75,j)
r(51,j)=dyn(76,j)
r(52,j)=dyn(77,j)
r(53,j)=dyn(78,j)
r(54,j)=dyn(79,j)
r(266,j)=dyn(80,j)
r(267,j)=dyn(81,j)
r(55,j)=dyn(82,j)
r(56,j)=dyn(83,j)
r(57,j)=dyn(84,j)
r(58,j)=dyn(85,j)
r(268,j)=dyn(86,j)
r(269,j)=dyn(87,j)
r(59,j)=dyn(88,j)
r(60,j)=dyn(89,j)
r(61,j)=dyn(90,j)
r(62,j)=dyn(91,j)
r(270,j)=dyn(92,j)
r(271,j)=dyn(93,j)
r(63,j)=dyn(94,j)
r(64,j)=dyn(95,j)
r(65,j)=dyn(96,j)
r(66,j)=dyn(97,j)
```

```
r(272,j)=dyn(98,j)
r(273,j)=dyn(99,j)
r(67,j)=dyn(100,j)
r(68,j)=dyn(101,j)
r(69,j)=dyn(102,j)
r(70,j)=dyn(103,j)
r(274,j)=dyn(104,j)
r(275,j)=dyn(105,j)
r(71,j)=dyn(106,j)
r(72,j)=dyn(107,j)
r(73,j)=dyn(108,j)
r(74,j)=dyn(109,j)
r(276,j)=dyn(110,j)
r(277,j)=dyn(111,j)
r(75,j)=dyn(112,j)
r(76,j)=dyn(113,j)
r(77,j)=dyn(114,j)
r(78,j)=dyn(115,j)
r(278,j)=dyn(116,j)
r(279,j)=dyn(117,j)
r(79,j)=dyn(118,j)
r(80,j)=dyn(119,j)
r(81,j)=dyn(120,j)
r(82,j)=dyn(121,j)
r(280,j)=dyn(122,j)
r(281,j)=dyn(123,j)
r(83,j)=dyn(124,j)
r(84,j)=dyn(125,j)
r(85,j)=dyn(126,j)
r(86,j)=dyn(127,j)
r(282,j)=dyn(128,j)
r(283,j)=dyn(129,j)
r(87,j)=dyn(130,j)
r(88,j)=dyn(131,j)
r(89,j)=dyn(132,j)
r(90,j)=dyn(133,j)
r(284,j)=dyn(134,j)
r(285,j)=dyn(135,j)
r(91,j)=dyn(136,j)
r(92,j)=dyn(137,j)
r(93,j)=dyn(138,j)
r(94,j)=dyn(139,j)
r(286,j)=dyn(140,j)
r(287,j)=dyn(141,j)
r(95,j)=dyn(142,j)
r(96,j)=dyn(143,j)
r(97,j)=dyn(144,j)
r(98,j)=dyn(145,j)
r(288,j)=dyn(146,j)
r(289,j)=dyn(147,j)
r(99,j)=dyn(148,j)
r(100,j)=dyn(149,j)
r(101,j)=dyn(150,j)
r(102,j)=dyn(151,j)
r(290,j)=dyn(152,j)
r(291,j)=dyn(153,j)
r(103,j)=dyn(154,j)
r(104,j)=dyn(155,j)
r(105,j)=dyn(156,j)
r(106,j)=dyn(157,j)
```

```
r(292,j)=dyn(158,j)
r(293,j)=dyn(159,j)
r(107,j)=dyn(160,j)
r(108,j)=dyn(161,j)
r(109,j)=dyn(162,j)
r(110,j)=dyn(163,j)
r(294,j)=dyn(164,j)
r(295,j)=dyn(165,j)
r(111,j)=dyn(166,j)
r(112,j)=dyn(167,j)
r(113,j)=dyn(168,j)
r(114,j)=dyn(169,j)
r(296,j)=dyn(170,j)
r(297,j)=dyn(171,j)
r(115,j)=dyn(172,j)
r(116,j)=dyn(173,j)
r(117,j)=dyn(174,j)
r(118,j)=dyn(175,j)
r(298,j)=dyn(176,j)
r(299,j)=dyn(177,j)
r(119,j)=dyn(178,j)
r(120,j)=dyn(179,j)
r(121,j)=dyn(180,j)
r(122,j)=dyn(181,j)
r(300,j)=dyn(182,j)
r(301,j)=dyn(183,j)
r(123,j)=dyn(184,j)
r(124,j)=dyn(185,j)
r(125,j)=dyn(186,j)
r(126,j)=dyn(187,j)
r(302,j)=dyn(188,j)
r(303,j)=dyn(189,j)
r(127,j)=dyn(190,j)
r(128,j)=dyn(191,j)
r(129,j)=dyn(192,j)
r(130,j)=dyn(193,j)
r(304,j)=dyn(194,j)
r(305,j)=dyn(195,j)
r(131,j)=dyn(196,j)
r(132,j)=dyn(197,j)
r(133,j)=dyn(198,j)
r(134,j)=dyn(199,j)
r(306,j)=dyn(200,j)
r(307,j)=dyn(201,j)
r(135,j)=dyn(202,j)
r(136,j)=dyn(203,j)
r(137,j)=dyn(204,j)
r(138,j)=dyn(205,j)
r(308,j)=dyn(206,j)
r(309,j)=dyn(207,j)
r(139,j)=dyn(208,j)
r(140,j)=dyn(209,j)
r(141,j)=dyn(210,j)
r(142,j)=dyn(211,j)
r(310,j)=dyn(212,j)
r(311,j)=dyn(213,j)
r(143,j)=dyn(214,j)
r(144,j)=dyn(215,j)
r(145,j)=dyn(216,j)
r(146,j)=dyn(217,j)
```

```
r(312,j)=dyn(218,j)
r(313,j)=dyn(219,j)
r(147,j)=dyn(220,j)
r(148,j)=dyn(221,j)
r(149,j)=dyn(222,j)
r(150,j)=dyn(223,j)
r(314,j)=dyn(224,j)
r(315,j)=dyn(225,j)
r(151,j)=dyn(226,j)
r(152,j)=dyn(227,j)
r(153,j)=dyn(228,j)
r(154,j)=dyn(229,j)
r(316,j)=dyn(230,j)
r(317,j)=dyn(231,j)
r(155,j)=dyn(232,j)
r(156,j)=dyn(233,j)
r(157,j)=dyn(234,j)
r(158,j)=dyn(235,j)
r(318,j)=dyn(236,j)
r(319,j)=dyn(237,j)
r(159,j)=dyn(238,j)
r(160,j)=dyn(239,j)
r(161,j)=dyn(240,j)
r(162,j)=dyn(241,j)
r(320,j)=dyn(242,j)
r(321,j)=dyn(243,j)
r(163,j)=dyn(244,j)
r(164,j)=dyn(245,j)
r(165,j)=dyn(246,j)
r(166,j)=dyn(247,j)
r(322,j)=dyn(248,j)
r(323,j)=dyn(249,j)
r(167,j)=dyn(250,j)
r(168,j)=dyn(251,j)
r(169,j)=dyn(252,j)
r(170,j)=dyn(253,j)
r(324,j)=dyn(254,j)
r(325,j)=dyn(255,j)
r(171,j)=dyn(256,j)
r(172,j)=dyn(257,j)
r(173,j)=dyn(258,j)
r(174,j)=dyn(259,j)
r(326,j)=dyn(260,j)
r(327,j)=dyn(261,j)
r(175,j)=dyn(262,j)
r(176,j)=dyn(263,j)
r(177,j)=dyn(264,j)
r(178,j)=dyn(265,j)
r(328,j)=dyn(266,j)
r(329,j)=dyn(267,j)
r(179,j)=dyn(268,j)
r(180,j)=dyn(269,j)
r(181,j)=dyn(270,j)
r(182,j)=dyn(271,j)
r(330,j)=dyn(272,j)
r(331,j)=dyn(273,j)
r(183,j)=dyn(274,j)
r(184,j)=dyn(275,j)
r(185,j)=dyn(276,j)
r(186,j)=dyn(277,j)
```

```
r(332,j)=dyn(278,j)
r(333,j)=dyn(279,j)
r(187,j)=dyn(280,j)
r(188,j)=dyn(281,j)
r(189,j)=dyn(282,j)
r(190,j)=dyn(283,j)
r(334,j)=dyn(284,j)
r(335,j)=dyn(285,j)
r(191,j)=dyn(286,j)
r(192,j)=dyn(287,j)
r(193,j)=dyn(288,j)
r(194,j)=dyn(289,j)
r(336,j)=dyn(290,j)
r(337,j)=dyn(291,j)
r(195,j)=dyn(292,j)
r(196,j)=dyn(293,j)
r(197,j)=dyn(294,j)
r(198,j)=dyn(295,j)
r(338,j)=dyn(296,j)
r(339,j)=dyn(297,j)
r(199,j)=dyn(298,j)
r(200,j)=dyn(299,j)
r(201,j)=dyn(300,j)
r(202,j)=dyn(301,j)
r(340,j)=dyn(302,j)
r(341,j)=dyn(303,j)
r(203,j)=dyn(304,j)
r(204,j)=dyn(305,j)
r(205,j)=dyn(306,j)
r(206,j)=dyn(307,j)
r(342,j)=dyn(308,j)
r(343,j)=dyn(309,j)
r(207,j)=dyn(310,j)
r(208,j)=dyn(311,j)
r(209,j)=dyn(312,j)
r(210,j)=dyn(313,j)
r(344,j)=dyn(314,j)
r(345,j)=dyn(315,j)
r(211,j)=dyn(316,j)
r(212,j)=dyn(317,j)
r(213,j)=dyn(318,j)
r(214,j)=dyn(319,j)
r(346,j)=dyn(320,j)
r(347,j)=dyn(321,j)
r(215,j)=dyn(322,j)
r(216,j)=dyn(323,j)
r(217,j)=dyn(324,j)
r(218,j)=dyn(325,j)
r(348,j)=dyn(326,j)
r(349,j)=dyn(327,j)
r(219,j)=dyn(328,j)
r(220,j)=dyn(329,j)
r(221,j)=dyn(330,j)
r(222,j)=dyn(331,j)
r(350,j)=dyn(332,j)
r(351,j)=dyn(333,j)
r(223,j)=dyn(334,j)
r(224,j)=dyn(335,j)
r(225,j)=dyn(336,j)
r(226,j)=dyn(337,j)
```

```
      r(352,j)=dyn(338,j)
      r(353,j)=dyn(339,j)
      r(227,j)=dyn(340,j)
      r(228,j)=dyn(341,j)
      r(229,j)=dyn(342,j)
      r(230,j)=dyn(343,j)
      r(354,j)=dyn(344,j)
      r(355,j)=dyn(345,j)
      r(231,j)=dyn(346,j)
      r(232,j)=dyn(347,j)
      r(233,j)=dyn(348,j)
      r(234,j)=dyn(349,j)
      r(356,j)=dyn(350,j)
      r(357,j)=dyn(351,j)
      r(235,j)=dyn(352,j)
      r(236,j)=dyn(353,j)
      r(237,j)=dyn(354,j)
      r(238,j)=dyn(355,j)
      r(358,j)=dyn(356,j)
      r(359,j)=dyn(357,j)
      r(239,j)=dyn(358,j)
      r(240,j)=dyn(359,j)
      r(360,j)=dyn(360,j)
      r(361,j)=dyn(361,j)
      end do
c
c     REARRANGE COLUMNS SO INTERNAL DISPLACEMENTS COME FIRST
      do i=1,361
      g(i,1)=r(i,1)
      g(i,2)=r(i,2)
      g(i,241)=r(i,3)
      g(i,3)=r(i,4)
      g(i,4)=r(i,5)
      g(i,5)=r(i,6)
      g(i,6)=r(i,7)
      g(i,242)=r(i,8)
      g(i,243)=r(i,9)
      g(i,7)=r(i,10)
      g(i,8)=r(i,11)
      g(i,9)=r(i,12)
      g(i,10)=r(i,13)
      g(i,244)=r(i,14)
      g(i,245)=r(i,15)
      g(i,11)=r(i,16)
      g(i,12)=r(i,17)
      g(i,13)=r(i,18)
      g(i,14)=r(i,19)
      g(i,246)=r(i,20)
      g(i,247)=r(i,21)
      g(i,15)=r(i,22)
      g(i,16)=r(i,23)
      g(i,17)=r(i,24)
      g(i,18)=r(i,25)
      g(i,248)=r(i,26)
      g(i,249)=r(i,27)
      g(i,19)=r(i,28)
      g(i,20)=r(i,29)
      g(i,21)=r(i,30)
      g(i,22)=r(i,31)
      g(i,250)=r(i,32)
```

```
g(i,251)=r(i,33)
g(i,23)=r(i,34)
g(i,24)=r(i,35)
g(i,25)=r(i,36)
g(i,26)=r(i,37)
g(i,252)=r(i,38)
g(i,253)=r(i,39)
g(i,27)=r(i,40)
g(i,28)=r(i,41)
g(i,29)=r(i,42)
g(i,30)=r(i,43)
g(i,254)=r(i,44)
g(i,255)=r(i,45)
g(i,31)=r(i,46)
g(i,32)=r(i,47)
g(i,33)=r(i,48)
g(i,34)=r(i,49)
g(i,256)=r(i,50)
g(i,257)=r(i,51)
g(i,35)=r(i,52)
g(i,36)=r(i,53)
g(i,37)=r(i,54)
g(i,38)=r(i,55)
g(i,258)=r(i,56)
g(i,259)=r(i,57)
g(i,39)=r(i,58)
g(i,40)=r(i,59)
g(i,41)=r(i,60)
g(i,42)=r(i,61)
g(i,260)=r(i,62)
g(i,261)=r(i,63)
g(i,43)=r(i,64)
g(i,44)=r(i,65)
g(i,45)=r(i,66)
g(i,46)=r(i,67)
g(i,262)=r(i,68)
g(i,263)=r(i,69)
g(i,47)=r(i,70)
g(i,48)=r(i,71)
g(i,49)=r(i,72)
g(i,50)=r(i,73)
g(i,264)=r(i,74)
g(i,265)=r(i,75)
g(i,51)=r(i,76)
g(i,52)=r(i,77)
g(i,53)=r(i,78)
g(i,54)=r(i,79)
g(i,266)=r(i,80)
g(i,267)=r(i,81)
g(i,55)=r(i,82)
g(i,56)=r(i,83)
g(i,57)=r(i,84)
g(i,58)=r(i,85)
g(i,268)=r(i,86)
g(i,269)=r(i,87)
g(i,59)=r(i,88)
g(i,60)=r(i,89)
g(i,61)=r(i,90)
g(i,62)=r(i,91)
g(i,270)=r(i,92)
```

```
g(i,271)=r(i,93)
g(i,63)=r(i,94)
g(i,64)=r(i,95)
g(i,65)=r(i,96)
g(i,66)=r(i,97)
g(i,272)=r(i,98)
g(i,273)=r(i,99)
g(i,67)=r(i,100)
g(i,68)=r(i,101)
g(i,69)=r(i,102)
g(i,70)=r(i,103)
g(i,274)=r(i,104)
g(i,275)=r(i,105)
g(i,71)=r(i,106)
g(i,72)=r(i,107)
g(i,73)=r(i,108)
g(i,74)=r(i,109)
g(i,276)=r(i,110)
g(i,277)=r(i,111)
g(i,75)=r(i,112)
g(i,76)=r(i,113)
g(i,77)=r(i,114)
g(i,78)=r(i,115)
g(i,278)=r(i,116)
g(i,279)=r(i,117)
g(i,79)=r(i,118)
g(i,80)=r(i,119)
g(i,81)=r(i,120)
g(i,82)=r(i,121)
g(i,280)=r(i,122)
g(i,281)=r(i,123)
g(i,83)=r(i,124)
g(i,84)=r(i,125)
g(i,85)=r(i,126)
g(i,86)=r(i,127)
g(i,282)=r(i,128)
g(i,283)=r(i,129)
g(i,87)=r(i,130)
g(i,88)=r(i,131)
g(i,89)=r(i,132)
g(i,90)=r(i,133)
g(i,284)=r(i,134)
g(i,285)=r(i,135)
g(i,91)=r(i,136)
g(i,92)=r(i,137)
g(i,93)=r(i,138)
g(i,94)=r(i,139)
g(i,286)=r(i,140)
g(i,287)=r(i,141)
g(i,95)=r(i,142)
g(i,96)=r(i,143)
g(i,97)=r(i,144)
g(i,98)=r(i,145)
g(i,288)=r(i,146)
g(i,289)=r(i,147)
g(i,99)=r(i,148)
g(i,100)=r(i,149)
g(i,101)=r(i,150)
g(i,102)=r(i,151)
g(i,290)=r(i,152)
```

```
g(i,291)=r(i,153)
g(i,103)=r(i,154)
g(i,104)=r(i,155)
g(i,105)=r(i,156)
g(i,106)=r(i,157)
g(i,292)=r(i,158)
g(i,293)=r(i,159)
g(i,107)=r(i,160)
g(i,108)=r(i,161)
g(i,109)=r(i,162)
g(i,110)=r(i,163)
g(i,294)=r(i,164)
g(i,295)=r(i,165)
g(i,111)=r(i,166)
g(i,112)=r(i,167)
g(i,113)=r(i,168)
g(i,114)=r(i,169)
g(i,296)=r(i,170)
g(i,297)=r(i,171)
g(i,115)=r(i,172)
g(i,116)=r(i,173)
g(i,117)=r(i,174)
g(i,118)=r(i,175)
g(i,298)=r(i,176)
g(i,299)=r(i,177)
g(i,119)=r(i,178)
g(i,120)=r(i,179)
g(i,121)=r(i,180)
g(i,122)=r(i,181)
g(i,300)=r(i,182)
g(i,301)=r(i,183)
g(i,123)=r(i,184)
g(i,124)=r(i,185)
g(i,125)=r(i,186)
g(i,126)=r(i,187)
g(i,302)=r(i,188)
g(i,303)=r(i,189)
g(i,127)=r(i,190)
g(i,128)=r(i,191)
g(i,129)=r(i,192)
g(i,130)=r(i,193)
g(i,304)=r(i,194)
g(i,305)=r(i,195)
g(i,131)=r(i,196)
g(i,132)=r(i,197)
g(i,133)=r(i,198)
g(i,134)=r(i,199)
g(i,306)=r(i,200)
g(i,307)=r(i,201)
g(i,135)=r(i,202)
g(i,136)=r(i,203)
g(i,137)=r(i,204)
g(i,138)=r(i,205)
g(i,308)=r(i,206)
g(i,309)=r(i,207)
g(i,139)=r(i,208)
g(i,140)=r(i,209)
g(i,141)=r(i,210)
g(i,142)=r(i,211)
g(i,310)=r(i,212)
```

```
g(i,311)=r(i,213)
g(i,143)=r(i,214)
g(i,144)=r(i,215)
g(i,145)=r(i,216)
g(i,146)=r(i,217)
g(i,312)=r(i,218)
g(i,313)=r(i,219)
g(i,147)=r(i,220)
g(i,148)=r(i,221)
g(i,149)=r(i,222)
g(i,150)=r(i,223)
g(i,314)=r(i,224)
g(i,315)=r(i,225)
g(i,151)=r(i,226)
g(i,152)=r(i,227)
g(i,153)=r(i,228)
g(i,154)=r(i,229)
g(i,316)=r(i,230)
g(i,317)=r(i,231)
g(i,155)=r(i,232)
g(i,156)=r(i,233)
g(i,157)=r(i,234)
g(i,158)=r(i,235)
g(i,318)=r(i,236)
g(i,319)=r(i,237)
g(i,159)=r(i,238)
g(i,160)=r(i,239)
g(i,161)=r(i,240)
g(i,162)=r(i,241)
g(i,320)=r(i,242)
g(i,321)=r(i,243)
g(i,163)=r(i,244)
g(i,164)=r(i,245)
g(i,165)=r(i,246)
g(i,166)=r(i,247)
g(i,322)=r(i,248)
g(i,323)=r(i,249)
g(i,167)=r(i,250)
g(i,168)=r(i,251)
g(i,169)=r(i,252)
g(i,170)=r(i,253)
g(i,324)=r(i,254)
g(i,325)=r(i,255)
g(i,171)=r(i,256)
g(i,172)=r(i,257)
g(i,173)=r(i,258)
g(i,174)=r(i,259)
g(i,326)=r(i,260)
g(i,327)=r(i,261)
g(i,175)=r(i,262)
g(i,176)=r(i,263)
g(i,177)=r(i,264)
g(i,178)=r(i,265)
g(i,328)=r(i,266)
g(i,329)=r(i,267)
g(i,179)=r(i,268)
g(i,180)=r(i,269)
g(i,181)=r(i,270)
g(i,182)=r(i,271)
g(i,330)=r(i,272)
```

```
g(i,331)=r(i,273)
g(i,183)=r(i,274)
g(i,184)=r(i,275)
g(i,185)=r(i,276)
g(i,186)=r(i,277)
g(i,332)=r(i,278)
g(i,333)=r(i,279)
g(i,187)=r(i,280)
g(i,188)=r(i,281)
g(i,189)=r(i,282)
g(i,190)=r(i,283)
g(i,334)=r(i,284)
g(i,335)=r(i,285)
g(i,191)=r(i,286)
g(i,192)=r(i,287)
g(i,193)=r(i,288)
g(i,194)=r(i,289)
g(i,336)=r(i,290)
g(i,337)=r(i,291)
g(i,195)=r(i,292)
g(i,196)=r(i,293)
g(i,197)=r(i,294)
g(i,198)=r(i,295)
g(i,338)=r(i,296)
g(i,339)=r(i,297)
g(i,199)=r(i,298)
g(i,200)=r(i,299)
g(i,201)=r(i,300)
g(i,202)=r(i,301)
g(i,340)=r(i,302)
g(i,341)=r(i,303)
g(i,203)=r(i,304)
g(i,204)=r(i,305)
g(i,205)=r(i,306)
g(i,206)=r(i,307)
g(i,342)=r(i,308)
g(i,343)=r(i,309)
g(i,207)=r(i,310)
g(i,208)=r(i,311)
g(i,209)=r(i,312)
g(i,210)=r(i,313)
g(i,344)=r(i,314)
g(i,345)=r(i,315)
g(i,211)=r(i,316)
g(i,212)=r(i,317)
g(i,213)=r(i,318)
g(i,214)=r(i,319)
g(i,346)=r(i,320)
g(i,347)=r(i,321)
g(i,215)=r(i,322)
g(i,216)=r(i,323)
g(i,217)=r(i,324)
g(i,218)=r(i,325)
g(i,348)=r(i,326)
g(i,349)=r(i,327)
g(i,219)=r(i,328)
g(i,220)=r(i,329)
g(i,221)=r(i,330)
g(i,222)=r(i,331)
g(i,350)=r(i,332)
```

```
      g(i,351)=r(i,333)
      g(i,223)=r(i,334)
      g(i,224)=r(i,335)
      g(i,225)=r(i,336)
      g(i,226)=r(i,337)
      g(i,352)=r(i,338)
      g(i,353)=r(i,339)
      g(i,227)=r(i,340)
      g(i,228)=r(i,341)
      g(i,229)=r(i,342)
      g(i,230)=r(i,343)
      g(i,354)=r(i,344)
      g(i,355)=r(i,345)
      g(i,231)=r(i,346)
      g(i,232)=r(i,347)
      g(i,233)=r(i,348)
      g(i,234)=r(i,349)
      g(i,356)=r(i,350)
      g(i,357)=r(i,351)
      g(i,235)=r(i,352)
      g(i,236)=r(i,353)
      g(i,237)=r(i,354)
      g(i,238)=r(i,355)
      g(i,358)=r(i,356)
      g(i,359)=r(i,357)
      g(i,239)=r(i,358)
      g(i,240)=r(i,359)
      g(i,360)=r(i,360)
      g(i,361)=r(i,361)
      end do
c
c     ASSEMBLE TRANSFORMATION MATRIX, t(361,361)
C     WHICH CONVERTS SURFACE DOFs TO NORMAL AND TANGENTIAL
c
c     Upper left corner
      do i=1,240
         do j=1,240
            t(i,j)=0.0
         end do
         t(i,i)=1.0
      end do
c
c     Upper right corner
      do i=1,240
         do j=241,361
            t(i,j)=0.0
         end do
      end do
c
c     Lower left corner
      do i=241,361
         do j=1,240
            t(i,j)=0.0
         end do
      end do
c
c     Lower right corner
      do i=241,361
         do j=241,361
            t(i,j)=0.0
```

```
         end do
      end do
      t(241,241)=-1.0
      t(360,360)=1.0
      t(361,361)=1.0
c
c     READ IN SURFACE NODE COORDINATES AND COMPUTE SIN THETA AND COS THETA
c     FOR ALL BUT FIRST AND LAST SURFACE NODES
c     NOTE THETA IS THE POLAR ANGLE
      do i=1,59
         read(5,'(2f10.8)') xx,yy
         rr=sqrt(xx**2+yy**2)
         ssin(i)=xx/rr
         ccos(i)=yy/rr
      end do
c
      k=241
      do i=242,358,2
         t(i,i)=ssin(i-k)
         t(i,i+1)=-ccos(i-k)
         t(i+1,i)=ccos(i-k)
         t(i+1,i+1)=ssin(i-k)
         k=k+1
      end do
c
c     ASSEMBLE tt(361,361) = TRANSPOSE OF t(361,361)
      do i=1,361
         do j=1,361
            tt(j,i)=t(i,j)
         end do
      end do
c
c     MULTIPLY g BY t
      do i=1,361
         do j=1,361
               sum=0.0
               do k=1,361
                  sum=sum+g(i,k)*t(k,j)
               end do
               pint(i,j)=sum
         end do
      end do
c
      do i=1,361
         do j=1,361
               sum=0.0
               do k=1,361
                  sum=sum+tt(i,k)*pint(k,j)
               end do
               prod(i,j)=sum
         end do
      end do
c
c     TEST SYMMETRY OF TRANSFORMED FULL DYNAMICAL MATRIX
c     nerr=0
c     do i=1,361
c        do j=i,361
c           if (prod(j,i).ne.prod(i,j)) then
c              write (*,*)'Symm error in transf full dyn mtrx at (i,j)=',i,j
c              nerr=nerr+1
```

```
c             end if
c          end do
c       end do
c       write (*,*) 'Number of symmetry errors in transf full dyn mtrx =',nerr
c       if (nerr.gt.0) then
c          STOP
c       end if
c
c       REARRANGE ROWS SO ZERO (TANGENTIAL) SURFACE FORCES COME FIRST
        do i=1,240
           do j=1,361
                s(i,j)=prod(i,j)
           end do
        end do
        do j=1,361
        s(300,j)=prod(241,j)
        s(301,j)=prod(242,j)
        s(241,j)=prod(243,j)
        s(302,j)=prod(244,j)
        s(242,j)=prod(245,j)
        s(303,j)=prod(246,j)
        s(243,j)=prod(247,j)
        s(304,j)=prod(248,j)
        s(244,j)=prod(249,j)
        s(305,j)=prod(250,j)
        s(245,j)=prod(251,j)
        s(306,j)=prod(252,j)
        s(246,j)=prod(253,j)
        s(307,j)=prod(254,j)
        s(247,j)=prod(255,j)
        s(308,j)=prod(256,j)
        s(248,j)=prod(257,j)
        s(309,j)=prod(258,j)
        s(249,j)=prod(259,j)
        s(310,j)=prod(260,j)
        s(250,j)=prod(261,j)
        s(311,j)=prod(262,j)
        s(251,j)=prod(263,j)
        s(312,j)=prod(264,j)
        s(252,j)=prod(265,j)
        s(313,j)=prod(266,j)
        s(253,j)=prod(267,j)
        s(314,j)=prod(268,j)
        s(254,j)=prod(269,j)
        s(315,j)=prod(270,j)
        s(255,j)=prod(271,j)
        s(316,j)=prod(272,j)
        s(256,j)=prod(273,j)
        s(317,j)=prod(274,j)
        s(257,j)=prod(275,j)
        s(318,j)=prod(276,j)
        s(258,j)=prod(277,j)
        s(319,j)=prod(278,j)
        s(259,j)=prod(279,j)
        s(320,j)=prod(280,j)
        s(260,j)=prod(281,j)
        s(321,j)=prod(282,j)
        s(261,j)=prod(283,j)
        s(322,j)=prod(284,j)
        s(262,j)=prod(285,j)
```

```
s(323,j)=prod(286,j)
s(263,j)=prod(287,j)
s(324,j)=prod(288,j)
s(264,j)=prod(289,j)
s(325,j)=prod(290,j)
s(265,j)=prod(291,j)
s(326,j)=prod(292,j)
s(266,j)=prod(293,j)
s(327,j)=prod(294,j)
s(267,j)=prod(295,j)
s(328,j)=prod(296,j)
s(268,j)=prod(297,j)
s(329,j)=prod(298,j)
s(269,j)=prod(299,j)
s(330,j)=prod(300,j)
s(270,j)=prod(301,j)
s(331,j)=prod(302,j)
s(271,j)=prod(303,j)
s(332,j)=prod(304,j)
s(272,j)=prod(305,j)
s(333,j)=prod(306,j)
s(273,j)=prod(307,j)
s(334,j)=prod(308,j)
s(274,j)=prod(309,j)
s(335,j)=prod(310,j)
s(275,j)=prod(311,j)
s(336,j)=prod(312,j)
s(276,j)=prod(313,j)
s(337,j)=prod(314,j)
s(277,j)=prod(315,j)
s(338,j)=prod(316,j)
s(278,j)=prod(317,j)
s(339,j)=prod(318,j)
s(279,j)=prod(319,j)
s(340,j)=prod(320,j)
s(280,j)=prod(321,j)
s(341,j)=prod(322,j)
s(281    =prod(323,j)
s(342,j)=prod(324,j)
s(282,j)=prod(325,j)
s(343,j)=prod(326,j)
s(283,j)=prod(327,j)
s(344,j)=prod(328,j)
s(284,j)=prod(329,j)
s(345,j)=prod(330,j)
s(285,j)=prod(331,j)
s(346,j)=prod(332,j)
s(286,j)=prod(333,j)
s(347,j)=prod(334,j)
s(287,j)=prod(335,j)
s(348,j)=prod(336,j)
s(288,j)=prod(337,j)
s(349,j)=prod(338,j)
s(289,j)=prod(339,j)
s(350,j)=prod(340,j)
s(290,j)=prod(341,j)
s(351,j)=prod(342,j)
s(291,j)=prod(343,j)
s(352,j)=prod(344,j)
s(292,j)=prod(345,j)
```

```
      s(353,j)=prod(346,j)
      s(293,j)=prod(347,j)
      s(354,j)=prod(348,j)
      s(294,j)=prod(349,j)
      s(355,j)=prod(350,j)
      s(295,j)=prod(351,j)
      s(356,j)=prod(352,j)
      s(296,j)=prod(353,j)
      s(357,j)=prod(354,j)
      s(297,j)=prod(355,j)
      s(358,j)=prod(356,j)
      s(298,j)=prod(357,j)
      s(359,j)=prod(358,j)
      s(299,j)=prod(359,j)
      s(360,j)=prod(360,j)
      s(361,j)=prod(361,j)
      end do
c
c     REARRANGE COLUMNS SO TANGENTIAL SURFACE DISPLACEMENTS COME FIRST
      do i=1,361
         do j=1,240
              mat(i,j)=s(i,j)
         end do
      end do
      do i=1,361
      mat(i,300)=s(i,241)
      mat(i,301)=s(i,242)
      mat(i,241)=s(i,243)
      mat(i,302)=s(i,244)
      mat(i,242)=s(i,245)
      mat(i,303)=s(i,246)
      mat(i,243)=s(i,247)
      mat(i,304)=s(i,248)
      mat(i,244)=s(i,249)
      mat(i,305)=s(i,250)
      mat(i,245)=s(i,251)
      mat(i,306)=s(i,252)
      mat(i,246)=s(i,253)
      mat(i,307)=s(i,254)
      mat(i,247)=s(i,255)
      mat(i,308)=s(i,256)
      mat(i,248)=s(i,257)
      mat(i,309)=s(i,258)
      mat(i,249)=s(i,259)
      mat(i,310)=s(i,260)
      mat(i,250)=s(i,261)
      mat(i,311)=s(i,262)
      mat(i,251)=s(i,263)
      mat(i,312)=s(i,264)
      mat(i,252)=s(i,265)
      mat(i,313)=s(i,266)
      mat(i,253)=s(i,267)
      mat(i,314)=s(i,268)
      mat(i,254)=s(i,269)
      mat(i,315)=s(i,270)
      mat(i,255)=s(i,271)
      mat(i,316)=s(i,272)
      mat(i,256)=s(i,273)
      mat(i,317)=s(i,274)
      mat(i,257)=s(i,275)
```

```
mat(i,318)=s(i,276)
mat(i,258)=s(i,277)
mat(i,319)=s(i,278)
mat(i,259)=s(i,279)
mat(i,320)=s(i,280)
mat(i,260)=s(i,281)
mat(i,321)=s(i,282)
mat(i,261)=s(i,283)
mat(i,322)=s(i,284)
mat(i,262)=s(i,285)
mat(i,323)=s(i,286)
mat(i,263)=s(i,287)
mat(i,324)=s(i,288)
mat(i,264)=s(i,289)
mat(i,325)=s(i,290)
mat(i,265)=s(i,291)
mat(i,326)=s(i,292)
mat(i,266)=s(i,293)
mat(i,327)=s(i,294)
mat(i,267)=s(i,295)
mat(i,328)=s(i,296)
mat(i,268)=s(i,297)
mat(i,329)=s(i,298)
mat(i,269)=s(i,299)
mat(i,330)=s(i,300)
mat(i,270)=s(i,301)
mat(i,331)=s(i,302)
mat(i,271)=s(i,303)
mat(i,332)=s(i,304)
mat(i,272)=s(i,305)
mat(i,333)=s(i,306)
mat(i,273)=s(i,307)
mat(i,334)=s(i,308)
mat(i,274)=s(i,309)
mat(i,335)=s(i,310)
mat(i,275)=s(i,311)
mat(i,336)=s(i,312)
mat(i,276)=s(i,313)
mat(i,337)=s(i,314)
mat(i,277)=s(i,315)
mat(i,338)=s(i,316)
mat(i,278)=s(i,317)
mat(i,339)=s(i,318)
mat(i,279)=s(i,319)
mat(i,340)=s(i,320)
mat(i,280)=s(i,321)
mat(i,341)=s(i,322)
mat(i,281)=s(i,323)
mat(i,342)=s(i,324)
mat(i,282)=s(i,325)
mat(i,343)=s(i,326)
mat(i,283)=s(i,327)
mat(i,344)=s(i,328)
mat(i,284)=s(i,329)
mat(i,345)=s(i,330)
mat(i,285)=s(i,331)
mat(i,346)=s(i,332)
mat(i,286)=s(i,333)
mat(i,347)=s(i,334)
mat(i,287)=s(i,335)
```

```
      mat(i,348)=s(i,336)
      mat(i,288)=s(i,337)
      mat(i,349)=s(i,338)
      mat(i,289)=s(i,339)
      mat(i,350)=s(i,340)
      mat(i,290)=s(i,341)
      mat(i,351)=s(i,342)
      mat(i,291)=s(i,343)
      mat(i,352)=s(i,344)
      mat(i,292)=s(i,345)
      mat(i,353)=s(i,346)
      mat(i,293)=s(i,347)
      mat(i,354)=s(i,348)
      mat(i,294)=s(i,349)
      mat(i,355)=s(i,350)
      mat(i,295)=s(i,351)
      mat(i,356)=s(i,352)
      mat(i,296)=s(i,353)
      mat(i,357)=s(i,354)
      mat(i,297)=s(i,355)
      mat(i,358)=s(i,356)
      mat(i,298)=s(i,357)
      mat(i,359)=s(i,358)
      mat(i,299)=s(i,359)
      mat(i,360)=s(i,360)
      mat(i,361)=s(i,361)
      end do
c
c     PARTITION mat MATRIX INTO a,b,c AND d MATRICES
      do i=1,299
         do j=1,299
            a(i,j)=mat(i,j)
         end do
      end do
c
      do i=1,299
         do j=300,361
            b(i,j-299)=mat(i,j)
         end do
      end do
c
      do i=300,361
         do j=1,299
            c(i-299,j)=mat(i,j)
         end do
      end do
c
      do i=300,361
         do j=300,361
            d(i-299,j-299)=mat(i,j)
         end do
      end do
c
c     COMPUTE INVERSE OF MATRIX a
      n=299
      np=299
      do i=1,n
         do j=1,n
            y(i,j)=0.0
         end do
```

```
         y(i,i)=1.0
      end do
      call ludcmp(a,n,np,indx,d)
      do j=1,n
         call lubksb(a,n,np,indx,y(1,j))
      end do
c
c     MULTIPLY c BY y
      do i=1,62
         do j=1,299
              sum=0.0
              do k=1,299
                 sum=sum+c(i,k)*y(k,j)
              end do
              p(i,j)=sum
         end do
      end do
c
c     MULTIPLY p BY b
      do i=1,62
         do j=1,62
              sum=0.0
              do k=1,299
                 sum=sum+p(i,k)*b(k,j)
              end do
              pp(i,j)=sum
         end do
      end do
c
c     COMPUTE d-pp
      do i=1,62
         do j=1,62
              nn(i,j)=d(i,j)-pp(i,j)
         end do
      end do
c
C     TEST SYMMETRY OF REDUCED DYNAMICAL MATRIX
c     nerr=0
c     do i=1,62
c        do j=i,62
c           fracerr=Abs(nn(j,i)-nn(i,j))/Sqrt(Abs(nn(j,i)*nn(i,j)))
c           if (fracerr.ge.0.001) then
c              write (*,*)'Symmetry error in reduced dyn. mtrx at (i,j)=',i,j
c              nerr=nerr+1
c           end if
c        end do
c     end do
c     write (*,*) 'Number of symmetry errors in reduced dyn. matrix =',nerr
c     if (nerr.gt.0) then
c        STOP
c     end if
c
C     THIS COMPLETES THE REDUCTION TO NODAL SURF VEL DEG OF FREEDOM
C
c     STORE REDUCED DYNAMICAL MATRIX IN UNFORMATTED BINARY FORM
      do i=1,62
         write(4)(nn(i,j),j=1,62)
      end do
C
      end
```

```
c
c       END OF MAIN PROGRAM
c
        subroutine ludcmp(a,n,np,indx,d)
        parameter(nmax=299,tiny=1.0e-32)
        dimension a(299,299), indx(299), vv(299)
        d=1.0
        do i=1,n
           aamax=0.0
           do j=1,n
                if(abs(a(i,j)).gt.aamax) aamax=abs(a(i,j))
           end do
           if(aamax.eq.0.0) pause 'singular matrix.'
           vv(i)=1.0/aamax
        end do
c
        do j=1,n
           do i=1,j-1
                sum=a(i,j)
                do k=1,i-1
                   sum=sum-a(i,k)*a(k,j)
                end do
                a(i,j)=sum
           end do
           aamax=0.0
c
           do i=j,n
                sum=a(i,j)
                do k=1,j-1
                   sum=sum-a(i,k)*a(k,j)
                end do
                a(i,j)=sum
                dum=vv(i)*abs(sum)
                if(dum.ge.aamax) then
                   imax=i
                   aamax=dum
                end if
           end do
c
           if(j.ne.imax)then
                do k=1,n
                   dum=a(imax,k)
                   a(imax,k)=a(j,k)
                   a(j,k)=dum
                end do
                d=-d
                vv(imax)=vv(j)
           end if
           indx(j)=imax
           if(a(j,j).eq.0.) a(j,j)=tiny
           if(j.ne.n) then
                dum=1.0/a(j,j)
                do i=j+1,n
                   a(i,j)=a(i,j)*dum
                end do
           end if
        end do
        return
        end
c
```

```
c
      subroutine lubksb(a,n,np,indx,hi)
      dimension a(299,299),indx(299),hi(299)
      ii=0
      do i=1,n
         m=indx(i)
         sum=hi(m)
         hi(m)=hi(i)
         if(ii.ne.0) then
         do j=ii,i-1
              sum=sum-a(i,j)*hi(j)
         end do
         else if(sum.ne.0.) then
         ii=i
         end if
         hi(i)=sum
      end do
c
      do i=n,1,-1
         sum=hi(i)
         if(i.lt.n) then
              do j=i+1,n
                 sum=sum-a(i,j)*hi(j)
              end do
         end if
         hi(i)=sum/a(i,i)
      end do
      return
      end
```

APPENDIX C:

```
c       PROGRAM SPHMTRX: reads in reduced nodal dynamical matrix and produces
c       dynamical matrix in terms of first seven Legendre polynomials
c
        real pi,sum
        real xx,yy,rr,ccos(61)
        real Dnod(62,62),P(61,7),Cinv(7,7),Pt(7,61),Dsph(8,8)
C
C       OLD FILES
        open(unit=8,file='transform.sph',status='old')
        open(unit=9,file='reddynmtrx.dat',status='old',form='unformatted')
C
C       NEW FILES
        open(unit=12,file='Dsph.dat',status='new',form='unformatted')
        open(unit=13,file='Dsph1.lis',status='new')
        open(unit=14,file='Dsph2.lis',status='new')
C
        pi=acos(-1.0)
c
c       READ IN COORDINATES OF ALL EXCEPT FIRST AND LAST SURFACE NODES
C       AND COMPUTE COS OF EACH ANGLE.
c       Note first node is at theta=pi and last node is at theta=0 where
c       theta ia the polar angle
        do i=1,59
           read(8,'(2f10.8)') xx,yy
           rr=sqrt(xx**2+yy**2)
           ccos(i+1)=yy/rr
        end do
        ccos(1)=-1.0
        ccos(61)=1.0
c
c       UNFORMATTED READ IN REDUCED DYNAMICAL MATRIX Dnod(62,62)
        do i=1,62
           read(9)(Dnod(i,j),j=1,62)
        end do
c
c       COMPUTE NODAL DISPLACEMENT TRANSFORMATION MATRIX, P(61,7), USING THE
C       FIRST 7 LEGENDRE POLYNOMIALS EVALUATED AT COS(NODE3) THRU COS(NODE183)
C       Compute zeroth through sixth order Legendre polynomials
        do i=1,61
           c2=ccos(i)**2
           P(i,1)=1.0
           P(i,2)=ccos(i)
           P(i,3)=(3.0*c2-1.0)/2.0
           P(i,4)=((5.0*c2-3.0)*ccos(i))/2.0
           P(i,5)=((35.0*c2-30.0)*c2+3.0)/8.0
           P(i,6)=(((63.0*c2-70.0)*c2+15.0)*ccos(i))/8.0
           P(i,7)=(((231.0*c2-315.0)*c2+105.0)*c2-5.0)/16.0
        end do
c
c       COMPUTE TRANSPOSE OF NODAL DISPLACEMENT TRANSFORMATION MATRIX Pt(7,61)
        do i=1,7
           do j=1,61
              Pt(i,j)=P(j,i)
           end do
        end do
c
C       COMPUTE INVERSE OF LEGENDRE POLYNOMIAL NORMALIZATION CONSTANTS Cinv(7,7)
        do i=1,7
           do j=1,7
              Cinv(i,j)=0.0
```

```
      end do
   Cinv(i,i)=2.0/(2*(i-1)+1)
   end do
c
c  COMPUTE UPPER LEFT 7x7 BLOCK OF Dsph(8,8)=Cinv(7,7)*Pt(7,61)*Dnod(61,61)
   do i=1,7
      do j=1,7
         sum=0.0
         do k=1,7
            do m=1,61
               do n=1,61
                  sum=sum+Cinv(i,k)*Pt(k,m)*Dnod(m,n)*P(n,j)
               end do
            end do
         end do
         Dsph(i,j)=sum
      end do
   end do
c
c  COMPUTE UPPER RIGHT 7x1 BLOCK OF Dsph(8,8)=Cinv(7,7)*Pt(7,61)*Knod(61,1)
   do i=1,7
      sum=0.0
      do j=1,7
         do k=1,61
            sum=sum+Cinv(i,j)*Pt(j,k)*Dnod(k,62)
         end do
      end do
      Dsph(i,8)=sum
   end do
c
c  COMPUTE LOWER LEFT 1x7 BLOCK OF Dsph(8,8)=Knodt(1,61)*P(61,7)
   do j=1,7
      sum=0.0
      do k=1,61
         sum=sum+Dnod(62,k)*P(k,j)
      end do
      Dsph(8,j)=sum
   end do
c
c  BLOCKED CAPACITANCE IS UNCHANGED
   Dsph(8,8)=Dnod(62,62)
c
C  STORE Dsph(8,8)
   do i=1,8
      write(12)(Dsph(i,j),j=1,8)
      write(13,10)(Dsph(i,j),j=1,4)
      write(14,10)(Dsph(i,j),j=5,8)
   end do
10 format(3x,4e17.7,//)
   end
```

LIST OF REFERENCES

1. Varadan, V.V., and Varadan, V. K., "Configurations with Finite Numbers of Scatterers -- A Self-Consistent T-matrix Approach," *Journal of Acoustical Society of America*, v. 70, pp. 213-217, 13 March 1981.

2. Scandrett, C. L., and Canright, D. R., "Acoustic Interactions in Arrays of Spherical Elastic Shells," submitted to *Journal of Acoustical Society of America.*

3. Press, W. H., and others, *Numerical Recipes*, Cambridge University Press, 1986.

4. Zienkiewicz, O. C., and Morgan, K., *Finite Elements and Approximation*, John Wiley & Sons, Inc., 1983.

5. Naval Ocean Systems Center Report 1329, *Contributions of Individual Structural Modes to the Scattered Acoustic Field*, by G. W. Benthien, November 1989.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Superintendent Naval Postgraduate School ATTN: Chairman, Department of Physics (Code 61) Monterey, California 93943-5000	1
4.	Superintendent Naval Postgraduate School ATTN: Professor S. R. Baker (Code 61Ba) Monterey, California 93943-5000	8
5.	Superintendent Naval Postgraduate School ATTN: Professor D. R. Canright (Code 53Ca) Monterey, California 93943-5000	2
6.	Superintendent Naval Postgraduate School ATTN: Professor C. L. Scandrett (Code 53Sd) Monterey, California 93943-5000	2
7.	Superintendent Naval Postgraduate School ATTN: Professor O. B. Wilson (Code 61Wl) Monterey, California 93943-5000	2

8. Commanding Officer 1
Naval Ocean Systems Center
ATTN: George Benthien (Code 712)
San Diego, California 92152-5000

9. Commanding Officer 1
Naval Underwater Systems Center
ATTN: Roger Richards (Code 213)
New London, Connecticut 06320

10. Commanding Officer 1
Naval Research Lab-USRD
ATTN: Robert Timme
Orlando, Florida 32856

11. Commanding Officer 1
Naval Research Lab-USRD
ATTN: Michele McCollum
Orlando, Florida 32856

12. International Transducer Corporation 1
ATTN: Mr. Ender Kuntsal
869 Ward Drive
Santa Barbara, California 93111

13. institut superieur d'electronique du nord 1
41 Blvd Vauban
ATTN: Jean-Noel Decarpigny
59046 Lille, Cedex, France

14. institut superieur d'electronique du nord 1
41 Blvd Vauban
ATTN: Bernard Hamonic
59046 Lille, Cedex, France

15. Commanding Officer 2
Fleet Combat Training Center, Atlantic
ATTN: LT. Kathleen A. McLean
Virginia Beach, Virginia 23461-5200